DOCUMENT 155-91

DATA REDUCTION AND COMPUTER GROUP

GENERAL PRINCIPLES OF DIGITAL FILTERING AND A SURVEY OF FILTERS IN CURRENT RANGE USE

WHITE SANDS MISSILE RANGE
KWAJALEIN MISSILE RANGE
YUMA PROVING GROUND
DUGWAY PROVING GROUND
ELECTRONIC PROVING GROUND

ATLANTIC FLEET WEAPONS TRAINING FACILITY
NAVAL AIR WARFARE CENTER WEAPONS DIVISION
NAVAL AIR WARFARE CENTER AIRCRAFT DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION NEWPORT

30TH SPACE WING
45TH SPACE WING
AIR FORCE FLIGHT TEST CENTER
AIR FORCE DEVELOPMENT TEST CENTER
AIR FORCE WEAPONS AND TACTICS CENTER
DETACHMENT 2, SPACE AND MISSILE SYSTEMS CENTER

DOCUMENT 155-91

GENERAL PRINCIPLES OF DIGITAL FILTERING AND A SURVEY OF FILTERS IN CURRENT RANGE USE

AUGUST 1991

Prepared by

Data Reduction and Computer Group
Range Commanders Council

Published by

Secretariat
Range Commanders Council
U.S. Army White Sands Missile Range
New Mexico 88002-5110

TABLE OF CONTENTS

APPENDIX A (CONT'D) Page

REFERENCES

LIST OF FIGURES

CHAPTER 1 - GENERAL PRINCIPLES OF DIGITAL FILTERING

CHAPTER 2 - FILTER ANALYSIS IN THE TIME DOMAIN

CHAPTER 3 - FILTER ANALYSIS AND DESIGN IN THE FREQUENCY DOMAIN

LIST OF FIGURES (Cont'd)

CHAPTER 4 - CATEGORIES OF DIGITAL FILTERS

CHAPTER 5 - FREQUENCY DOMAIN FILTER DESIGN

LIST OF FIGURES (Cont'd)

CHAPTER 7 - FILTERS IN CURRENT RANGE USE

LIST OF TABLES

CHAPTER 3 - FILTER ANALYSIS AND DESIGN IN THE FREQUENCY DOMAIN

CHAPTER 5 - FREQUENCY DOMAIN FILTER DESIGN

CHAPTER 6 - KALMAN FILTERS

LIST OF TABLES (Cont'd)

CHAPTER 1

GENERAL PRINCIPLES OF DIGITAL FILTERING

Data reduction includes an inherent requirement to filter measured data. The part of the measured values which is noise, that is, which does not characterize the phenomena being measured, must be removed so that the data expresses the characteristics of the system being tested. (A system is a device which interrelates the excitation and the response.) Often it is necessary to perform integration and differentiation based on measured data. The digital filter is a means by which these operations can be performed. Without data filtering, the measurements and the derived values from these measurements often depict physical phenomena that are virtually impossible. Aircraft and missiles are predictable, so it is possible to design filters which model their physical performance.

The application of digital filters is continually growing as more computers are placed in line in the measuring systems, as classical measurement systems are being upgraded, and as new measuring data concepts are being introduced. Video signal image enhancement, video signal transmission, the Global Positioning System, and aircraft onboard systems will necessitate the development of new digital filtering theories. Filters have many different functions; for example, filters are used in water purification to remove those particles which are undesirable. Filters are also used in coffee pots to keep the grounds separate from the liquid. Digital filters serve much the same purpose; they separate true data from errors which are added during the measurement process.

Signal processing filters were originally circuits with frequency selection characteristics. Some of the filters which developed as a result of this concept were the Butterworth, the Chebyshev, and the elliptic. These filters can be designed to meet some specific criteria regarding amplitude and phase response, but because filters of this type work on data frequency characteristics, they are not very useful in their original realizations on data recorded in digital form. There are time domain realizations which exhibit similar characteristics on time domain data. When digital realizations exist in software, a design error can be easily corrected; when they do not exist, however, roundoff error, storage requirements, and delays become important design factors.

Digital filters are not always derived from a frequency domain realization. Filters can be used directly with digital data, based on the statistical characteristics of the useful signal and noise. The Kalman filters belong to this class of

filters. Some of the principles used in filters of this type were developed by Gauss in his concept of weighted least squares. Although he used this concept in data analysis, it was not until the advent of digital computers that the full potential of this concept was realized. A few extensions have been added to Gauss' work, and many techniques have been developed based on his principles.

Related to digital filters are the terms smoother, filter, predictor, and wild point. These terms are included because the range surveys in appendix A include references to these terms. Given measurements in the time interval [0,t], a smoother produces estimates for times $t_s < t$, a filter produces estimates for times $t_f = t$, and a predictor produces estimates for times $t_p > t$. Thus, optimal estimates are obtained postflight by smoothers (using all recorded data) and are obtained in real time by filters (using all data up to the present). A wild point is a measured data value that fails to meet some predetermined statistical criteria. A "wild point application" of a digital filter is the replacement of a wild point.

A few of the current digital filtering applications by the Range Commanders Council (RCC) member ranges are

(1) in rocket firing to give position, velocity, and acceleration in real time for range safety information;

(2) in inertial navigation where navigation errors have to be estimated in real time for corrections to be applied;

(3) in radar and theodolite data processing to give trajectory estimates for position, velocity, and acceleration;

(4) in satellite trajectory determination;

(5) in processing airplane flight test data; and

(6) in reconstructing photographs or voices which have been transmitted digitally.

This document catalogues some types of digital filters widely used by the RCC member ranges, explores the principles behind these filters, and shows the design techniques for specific applications. Obviously, a rigorous, theoretical treatment of the subject cannot be given in this document, nor can all the details of design techniques. The intent is to give the reader a feel for the subject of digital filtering as defined by the members of the RCC Data Reduction and Computer Group. Noted are the appropriate bibliography references for any of the subject areas where more detail might be helpful to the reader.

1.1 Time Domain Measurement Data

Two representations can be used for discrete data in the time domain. The most commonly used definition for a sampled function is the sequence {f(n)} whose elements are

$$f(n)=f(t(n))=f(t)\Big|_{t=t(n)} \qquad (1\text{-}1)$$

where n is an index for the times at which samples are taken and f(t) is the continuous function from which the samples are taken. In further discussion in this paper, this representation will be used for measurement data. Another mathematical representation for a sampled function in the time domain, f*(t), is

$$f^*(t) = \sum_k f(t(k))\ \delta(t-t(k)) \qquad (1\text{-}2)$$

where the t(k)'s are the instants in time at which the function is sampled and $\delta(t)$ is the Dirac delta function or impulse function having the properties of

$$\delta(t-a) = \begin{cases} 0 & t \neq a \\ \infty & t = a \end{cases} \qquad (1\text{-}3)$$

$$\int_{a-\epsilon}^{a+\epsilon} \delta(t-a)\ dt = 1 \qquad \epsilon > 0 \qquad (1\text{-}4)$$

The above definition of $\delta(t)$ is from reference 47, page 15. Another common definition is

$$\delta(t) = 0,\ t \neq 0,$$
$$\delta(0) = \infty,\ \text{and}$$

$$\int_{-\epsilon}^{\epsilon} \delta(t)dt = 1. \qquad \epsilon > 0 \qquad (1\text{-}5)$$

As explained in reference 48, page 70, the function represents an imaginary rectangle whose base is 0, height is ∞, and area is 1.

The first representation allows filters to be analyzed using some of the calculus techniques of continuous functions and allows bridging of the gap between discrete and continuous analysis. In many current papers on this topic, the mathematical model for the measurements and the state of a system are given in terms of vectors. In this notation, a group of variables is

treated as one vector variable; hence, a compactness of notation is obtained, and very general filters can be discussed without becoming immersed in details. For the sake of completeness, this representation is mentioned here.

Consider a sequence of measurements which are represented in a form similar to equation (1-2). Instead of considering a sequence of real numbers, consider a sequence of vectors, {X(n)}. A particular element X(n) of this sequence consists of measurements made at time t(n) and should relate to the true state of the process from which the measurements were taken. For instance, X(n) could consist of measurements of azimuth, elevation, and range for some object in space. It is known that the sequence {X(n)} can be related, for example, to the position, velocity, and acceleration in the x, y, and z directions. Now assume that the true state of the process Y(t) can be represented by the differential equation

$$\frac{dY(t)}{dt} = F(Y(t),t) \tag{1-6}$$

where Y(t) is a vector composed of scalar functions y(t,1),..., y(t,N) and all the derivatives of each of these scalars; F(Y(t),t) is a vector function that is, in general, nonlinear and has as each element a function of all the elements in Y. Y(t) is known as the state vector and could consist of x, y, and z coordinate positions along with the first and second derivatives of these positions, for instance. Further assume that X(n), the measurement vector, is related to the state vector as

$$X(n) = G(Y(t(n)),t(n)) + N(n) \tag{1-7}$$

where G is the measurement function of the state variable and N(n) is the noise vector or the measurement error vector. In many applications, equation (1-6) constitutes a linear and homogeneous system, and the relationship between the measurement vector and the state vector is assumed linear. In this case, equations (1-6) and (1-7) reduce to

$$\frac{dY}{dt} = A(t)Y(t) \tag{1-8}$$

$$X(n) = M(n)Y(n)+N(n) \tag{1-9}$$

where A(t) is a matrix function of time and M(n) is a matrix known as the measurement matrix. In digital filtering, equations (1-6) and (1-7) are used to give some type of best estimate for Y(t).

In many cases, the measurements are made on certain elements of the state vector itself; that is, the measurement and state vector frequently represent the same test item. For example, an airplane's azimuth and elevation may be measured from several different stations to obtain its "true" azimuth and elevation. In this case, the matrix M(n) is diagonal and contains only 1s and 0s on the diagonal. Most future discussions will operate under this assumption. For ease of understanding, this document will consider the case where an attempt is made to estimate one variable from measurements made on that variable. The state differential equation will be implied rather than given explicitly.

1.2 Error Assumptions

In digital filter analysis, three types of errors are considered: (1) errors in the mathematical model, (2) random errors in the observations, and (3) unwanted discrete frequencies in the observations. The first type of error is caused by a lack of understanding of the physical system for which the estimate is to be made. The second error type is due to inaccurate measurements and is assumed to be normally distributed and unbiased. The third error type is caused by either the reception of noise along with the desired frequency signal or the intention to block out one or more bands of frequencies to suit a specific application. (If a digital filter is used for the second reason, then the third type of "error" is not really an error.)

The random errors are discussed next. Assume that {x(n)} is a sequence of measurements corrupted by noise {e(n)} and that the true value of the process is given by {Y(n)}. In all applications the noise is considered to be additive so that the equation relating the state variable to the measured variable at time t(n) is given by

$$x(n) = Y(n)+e(n). \tag{1-10}$$

It is often assumed that the noise is Gaussian with zero mean and uncorrelated with itself. The quantity which is most often used to characterize the noise is the variance given by equation

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} e(i)^2 \tag{1-11}$$

where N is the total number of measurements being considered. How much this variance is reduced by the filter may be used as the criterion for determining the filter characteristics. Caution must be exercised when using this criterion, because {e(i)} is a time sequence and thus, in most cases, correlated with itself.

Now consider errors of unwanted discrete frequencies in the observations. Again consider the relationship between the true value Y(n), the measurement x(n), and the error e(n) to be of the form in equation (1-10) with {e(n)} being a signal containing unwanted frequencies. In this case, it is assumed that {Y(n)} and {e(n)} are not in the same frequency band. This property is best illustrated using the power spectral density function, the quantity which indicates the extent that a particular range of frequencies present in the data. In the case when the frequency range of the signal and error are nonoverlapping, then the power spectral density function $G(\omega)$ would appear as shown in figure 1-1.

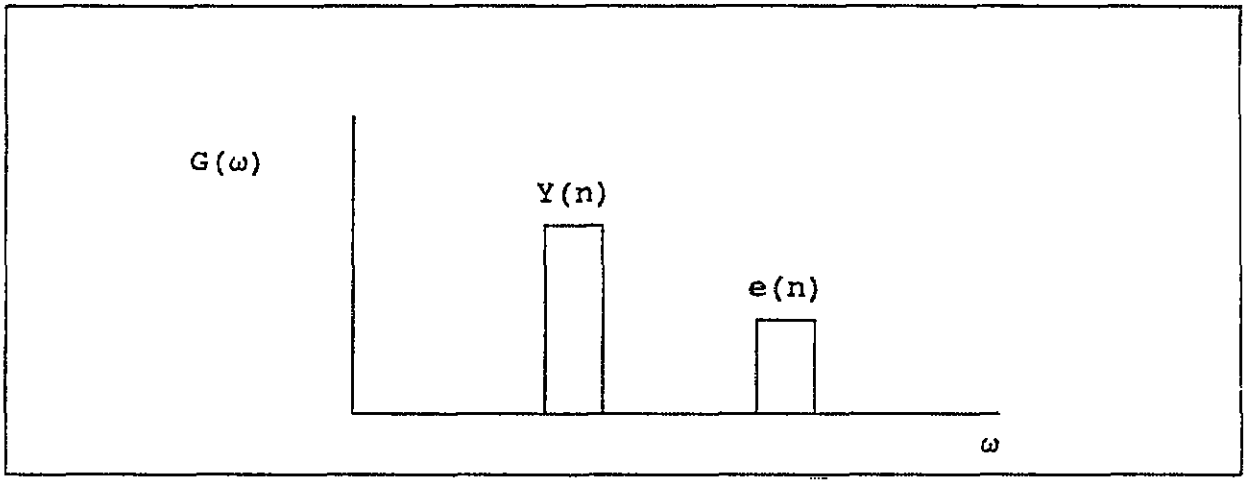

Figure 1-1. Graph of power spectrum $G(\omega)$ versus frequency ω, comparing true frequency signal {Y(n)} with noisy frequency signal {e(n)}.

In this case, a filter would be designed having a frequency response such that {Y(n)} would pass through unchanged, whereas {e(n)} would be suppressed as much as possible.

The power spectral density function for Gaussian white noise is a constant as shown in figure 1-2. When {e(n)} is of this form, the signal and noise are in overlapping frequency ranges; hence, part of the noise is treated as signal and is not suppressed. However, the desired signal is often of a low enough frequency and of a sufficient amplitude that most of the noise in this frequency range can be neglected.

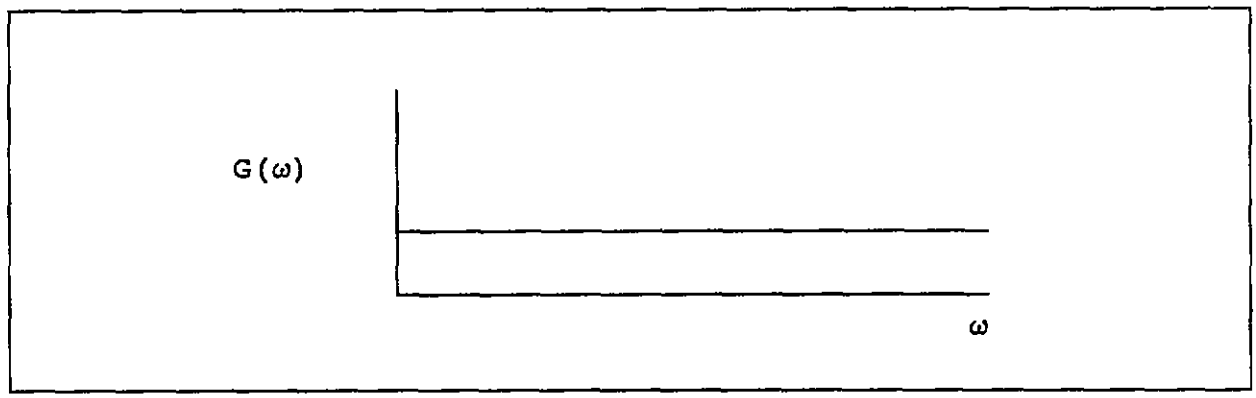

Figure 1-2. Power spectrum of Gaussian white noise.

1.3 Basic Purpose and Filter Use

Filters are used to ascertain information about a process from measurements made on that process when these measurements are corrupted by noise. Any of the following data should be incorporated into the filter if known:

(1) the differential equations describing the process,

(2) the statistics describing the noise, and

(3) the frequency range of the signals describing the process.

In many applications the differential equation describing the system is not known. In these cases, it is often assumed that over small enough time intervals the differential equation of the process is given by

$$\frac{d^k y}{dt^k} = 0 \qquad (1\text{-}12)$$

for some positive integer k.

The noise most often encountered in physical applications is Gaussian; however, often the variance is not known. When the variance of the noise is not known, a least squares criterion is often used to design a filter. The least squares criterion requires that

$$\sum_{n=0}^{N} (\hat{y}(n)-y(n))^2 = \text{a minimum} \qquad (1\text{-}13)$$

where $\{\hat{y}(n)\}$ is the sequence obtained by some type of fit to the data, which is used to give a filtered output. Combining equations (1-12) and (1-13) gives birth to the many filters which are based on least squares polynomial approximation. There are many variations of these filters used in a variety of applications in both real and nonreal time. Probably the widest application of these filters is found in the processing of tracking data. For example, in processing radar data, azimuth, elevation, and range are measured. From these measurements a raw x, y, and z are computed. The digital filter is used to obtain smoothed values for x, y, and z as well as the first and second derivatives for these quantities. Other uses for filters, based on least squares polynomials, include processing of theodolite tracking data and obtaining rates of climb and accelerations from altitude and airspeed data measured onboard an airplane. When the differential equation describing the process and the statistics of the noise are both known, then a Kalman filter can be used to estimate the process. The Kalman filter has found wide application in the fields of guidance and navigation. (The Kalman filter is discussed in chapter 6.)

When the signal describing the process lies in a certain frequency band, a filter is designed which will allow only the frequencies in this band to pass. (Passing frequencies of a certain band are discussed in chapter 4.) These filters are used for square-law detection, frequency-selective smoothing, phase and amplitude determination, and smoothing differentiation. Specific applications include processing of digitally transmitted photographs, medical records, and tracking data. The determination of the best filter for any particular application requires a careful analysis. The filter which is ultimately selected will depend on the physical system under analysis, the type of measurements being taken, and the constraints in computation time.

1.4 Linear Filters

Linear filters are filters which conform to

$$F(x+ay) = F(x)+aF(y), \qquad (1\text{-}14)$$

where F denotes the filtering operation, x and y denote inputs to the filter, and a is a constant. For such filters, the output at a particular time equals the input multiplied by a set of weights which are not a function of the output as given by the equation

$$y(k) = \sum_j h(k,j)x(k-j) \qquad (1\text{-}15)$$

The set $\{h(k,j)\}$ denotes the weight function described in paragraph 1.5. The advantage of linear filters is that they operate on signal and noise independently, so it is relatively easy to determine how they will treat both signal and noise.

Most of the filters in current use are linear; least squares moving-arc polynomials and most applications of frequency-constrained and Kalman filters fall in this category. (The least squares moving-arc polynomials filter is discussed in chapter 7.) If the system under consideration is nonlinear, then some type of linearizing scheme is used so that some of the nice properties of linearity can be retained.

1.5 Filter Weight Functions

To illustrate the idea of filter weight functions, consider a continuous linear filter having one input variable and one output variable. Such a filter is defined by

$$y(t) = \int_{-\infty}^{\infty} h(t,T)x(t-T)dT \qquad (1\text{-}16)$$

The integral in equation (1-16) is the well-known convolution integral. Here x(t) is the input, y(t) is the output, and h(t,T) is the filter weight function. The discrete analog to equation (1-16) is

$$y(j) = \Sigma_k h(j,k)x(j-k) \qquad (1\text{-}17)$$

where {x(j)} is the input sequence, {y(j)} is the output sequence, and {h(j,k)} is a set of weights corresponding to the weight function previously defined. The last equation characterizes the nonrecursive digital filter discussed in paragraph 1.8. The summation in equation (1-17) is known as the convolution sum. The filters, as defined, can be time varying; that is, the weight function can change with time. In the continuous case, if h(t,T) is a function of T only, then the filter is time invariant. Similarly, in the discrete case, if {h(j,k)} is a function of k only, then the digital filter is time invariant. The equation for a digital filter then becomes

$$y(j) = \Sigma_k h(k)x(j-k) \qquad (1\text{-}18)$$

A digital filter is time invariant when the system is being modeled by a constant coefficient differential equation, and the relationship between the state variables and the measurements does not vary with time. When measurements are made at equal increments in time, the set of weights for a time-invariant filter remains the same for each output point. Thus, time invariance is a desirable feature from a computational point of view.

1.6 General Purpose Filters

Many filters in use today are general purpose; that is, they are not restricted to a particular application. The most commonly used of these filters are the least squares polynomial filters. These filters are usually designed so that the degree and number of points to be used can be specified. Within this category there are many variations. For instance, a filter can be constrained so that it passes through the last filtered point and has continuous slope through that point. In addition, the specific implementation can vary; for example, the recursive sums or orthogonal polynomials are different implementations of the polynomial filters.

Most of the frequency-constrained filters mentioned in paragraph 1.4 are general-purpose filters. Any one of these filters can be used in a number of applications. Some uses of general-purpose filters were mentioned in paragraph 1.3. In many instances, these filters can be moved from one application to another without modification.

1.7 Special Purpose Filters

Certain filters are designed for a special mission. Once designed and checked out, these filters then ideally remain the same for the duration of the mission. Two important categories of special purpose filters are range safety and navigation system. An example of a special purpose range-safety filter is the QD filter, (see paragraph 7.7). This filter is suited for its mission because of the speed with which it can deliver smoothed position, velocity, and acceleration in real time. It also has the ability to do spike editing in real time. (Spike editing is defined as editing out the wild points or spikes.) For navigation filtering, special implementations of the Kalman filter are usually used. The Kalman filter is well suited for this application because of its ability to give optimal estimates of a system's state variables in time.

1.8 Nonrecursive Filters

In paragraph 1.5, a digital filter was defined in equation (1-17). For a filter defined in this way, each output point is a function only of the input and the weight function. Any filter for which the current output is not a function of previous outputs such as the one defined by equation (1-17) is said to be nonrecursive. Nonrecursive filters are analogous to open-loop filters in servomechanisms. Many implementations of the least squares polynomial filters and of the frequency-constrained filters fall into this category. Such filters are not subject to problems of instability; that is, an error occurring in a computed output point does not propagate into future output points.

If the right-hand side of equation (1-18) represents a finite sum, then it is called a finite impulse response (FIR) system or FIR filter. In FIR filters, if the right-hand side of equation (1-18) contains q terms, then the filter can have an impulse response at most q samples long. If the last impulse occurred more than q samples ago, the entire right-hand side of equation (1-18) is zero. A presentation of design methods using FIR filters, along with a summary of advantages and disadvantages of using this filter type, is given in chapter 5.

1.9 Recursive Filters

Recursive filters use past output values in computing the current output value. These filters take advantage of past computations in such a way that output values are used to yield information about previous input values to the filter. In this way, storage and computation time are saved. The general form of a recursive filter having one input and one output is

$$y(k) = \sum_{0}^{N} a(j)x(k-j) + \sum_{1}^{M} b(j)y(k-j) \qquad (1\text{-}19)$$

where $\{x(j)\}$ is the input, $\{y(j)\}$ is the output, and $\{a(j),b(j):j=1,\ldots\}$ is the set of weights. Some recursive filters (for example, Kalman and QD) use the previous output to predict a value for the current output and use the current measured input to correct the current output. Recursive filters are analogous to closed-loop filters in servomechanisms. Another commonly used name for such filters is infinite impulse response (IIR) filters or IIR systems. In IIR filters, all outputs y(k) will be influenced by all previous impulse responses a(j) in equation (1-19) regardless of how large k is. Each impulse response has an influence on an infinite number of terms y(k).

The savings in computation time and storage achieved by using recursive filters do not come without a price. If not properly designed, the filters can suffer from instability. Because each filter is computed from previous values $y(j), j<k$, an error induced in any one of these y(k)'s will have an effect in each future y(k). If such errors do not die out as k increases, then the filter is unstable and is not usable. Ways of analyzing the stability characteristics of filters are discussed in chapter 2. Another factor to consider in using recursive filters is the fact that they are not self starting. For the first few output points there will be no previous outputs to use in equation (1-19). Hence, some other method must be devised through which the initial y(k)'s are determined. Methods of choosing starting values depend, of course, on the particular application in

question and are discussed in references 1 and 42 and in chapter 7. A presentation of design methods using IIR filters, along with a summary of advantages and disadvantages, is given in chapter 5.

1.10 Filter Weights Construction

Again consider a nonrecursive, nontime-varying linear filter of the form

$$y(k) = \sum_j h(j,k)x(k-j) \qquad (1\text{-}20)$$

The problem in designing a filter is finding the weights $\{h(j,k)\}$ in such a way that some design criterion is satisfied. Some of the examples discussed in succeeding chapters are

(1) choosing weights by requiring that the span of $\{x(j)\}$ under consideration be fitted by a least squares polynomial,

(2) choosing the weights using Fourier methods by requiring that the frequency response of the filter fit an idealized frequency response function, and

(3) using the z-transform to derive filter weights from the transfer function of a given analog filter (see chapter 3).

When the filter is recursive, and thus takes the form

$$y(k) = \sum_{0}^{N} a(j)x(k-j) + \sum_{1}^{M} b(j)y(k-j) \qquad (1\text{-}21)$$

then there is a corresponding nonrecursive filter which is theoretically the same. In this case, the weights are either derived in terms of the corresponding weights for the nonrecursive case or are derived directly using z-transforms.

CHAPTER 2

FILTER ANALYSIS IN THE TIME DOMAIN

2.1 Filter Characteristics Analysis

Analysis of a filter in the time domain consists of determining the characteristics. These characteristics are

(1) Stability. With a unit impulse input to a recursive filter, the filter is unstable if the output oscillates or never dies down to zero amplitude or attenuates.

(2) Attenuation. With a nonzero input parameter, attenuation is the amount the filter has reduced its amplitude in the output.

(3) Time Lag. The amount of time it takes for a filter to supply a best estimate of an input parameter.

(4) Distortion. Attenuating or amplifying amplitudes of different frequencies by different amounts in an undesirable manner is called distortion. If the filter causes minimal distortion, it is a good fit or well-modeled.

To analyze these characteristics, the following techniques are frequently used:

(1) Autocorrelation

(2) Unit Impulse Response

(3) Variance Reduction Factor

(4) Simulation

(5) Monte Carlo Methods

(6) Residuals

2.2 Autocorrelation

When data are fed into a filter, the output data may be more (linearly) correlated in time than the input data; that is, series of output data values may be more correlated than their corresponding series of input data values. The type and amount of this additional correlation will depend on the filter weights which are used as data multipliers. For example, if the filter

frequency response has large side lobe effects and passes frequencies which were supposed to be suppressed, the resultant output will show these effects and they can be measured if the serial autocorrelation in the data is estimated. The serial autocorrelation, used in analyzing time lag, is computed by

$$r(k) = \frac{\sum_{i=1}^{N-k} (x(i)-\bar{x}(1))(x(i+k)-\bar{x}(2))}{\left(\sum_{i=1}^{N-k} (x(i)-\bar{x}(1))^2 \sum_{i=1}^{N-k} (x(i+k)-\bar{x}(2))^2 \right)^{1/2}} \qquad (2\text{-}1)$$

where x(1) and x(2) are the means of the first N-k and the last N-k data points and

N = number of points in the sample,
k = 0,1,2,..., M<N, the number of lags.

The value r(k) represents the (linear) correlation between the first and last N-k data values from a series of N points. This technique can be used in conjunction with simulation techniques by entering white noise into the filter and then measuring the autocorrelation of the output. The amount of correlation imposed on the data by the filter is thus estimated.

Caution must be used if an attempt is made to "smooth" the same data more than once in a sequential manner because of the correlation which may be imposed on the data by the first smoothing process. In cases where it is desirable to smooth position and then again smooth computing velocity, a thorough analysis should be made to ensure that autocorrelation imposed by the first smoothing process does not bias the estimates made by the second process.

2.3 Unit Impulse Response

For a continuous, linear, time-invariant system, a unit impulse response h(t) can be defined as the response of the system at time t because of a unit impulse at time 0, that is, because of $\delta(0)$. (Unit impulse is another term for impulse function (see page 1-3.)) A system is linear if the input $c_1f_1(t)+c_2f_2(t)$ produces an output $c_1g_1(t)+c_2g_2(t)$ for all $f_1(t)$ and $f_2(t)$, where inputs $f_1(t)$ and $f_2(t)$ produce outputs $g_1(t)$ and $g_2(t)$.

The input f(t) and output g(t) of the system are related by

$$g(t) = \int_{-\infty}^{\infty} h(\tau)f(t-\tau)d\tau \qquad (2\text{-}2)$$

where $h(\tau)$ is the unit impulse response. Thus, the output equals the convolution of the unit impulse response and the input.

For a discrete, linear, time-invariant system, the unit impulse response is the sequence $\{h_n\}$ where h_n is the response of the system at state n caused by a unit impulse at state 0. For the discrete case, the input-output relationship can be described as

$$y_n = \sum_{m=-\infty}^{\infty} h_m x_{n-m} \qquad (2\text{-}3)$$

where $\{x_n\}$ and $\{y_n\}$ are the input and output signals, and $\{h_n\}$ is the unit impulse response.

For either case, the output of a linear, time-invariant system is thus the convolution of its input and its unit impulse response. For this purpose, equation (2-3) will represent the system of interest, since digital filters are defined to be discrete, linear, time-invariant systems. As seen from equation (2-3), the unit impulse response can be regarded as a set of weights. Thus, from what was discussed in paragraph 1.5, the unit impulse response completely determines the characteristics of a digital filter. Reference can then be made either to the weights, given past data, in the averaging process or to the unit impulse response.

In reality, it is important to know how much a filter distorts a deterministic input. One way to determine the distortion is to analyze unit impulse response. The z-transform is a convenient tool for the analysis of the response of digital filters when the unit impulse response is known. The z-transform for a sequence $\{x_n\}$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n} \qquad (2\text{-}4)$$

Stated earlier in this section is that the output of the system can be found by convolving the input and unit impulse response. The output can be found easier through the use of the z-transform. An example taken from reference 50 demonstrating the use of the z-transform in determining the output signal or the unit impulse response is described next.

Let the input signal be {2 1} and the unit impulse response be {8 4 2 1}. Referring to equation (2-2), the Convolution Theorem, which states that the z-transform of the convolution of two functions is equal to the product of the z-transforms of those functions, is applied (see reference 23 or 50). The z-transforms of the input signal and unit impulse response are

$$2 + z^{-1} \quad \text{and} \quad 8 + 4z^{-1} + 2z^{-2} + z^{-3} \tag{2-5}$$

The product of those z-transforms is

$$16 + 16z^{-1} + 8z^{-2} + 4z^{-3} + z^{-4}. \tag{2-6}$$

This polynomial is the z-transform of the output signal. Hence, our output signal is {16 16 8 4 1}.

To find the impulse response, the input and output must be given. Then, the Convolution Theorem can be applied by dividing the z-transform of the output by the z-transform of the input and expanding the result as a polynomial in z^{-1}. The coefficients of this polynomial represent the impulse response. To find the inpulse response, the process described in the previous example is reversed. For any given filter, the unit impulse response, which returns to and remains at zero, is an indication of the filter's stability, time-lag characteristic, and amplitude attenuation.

Another means of specifying a filter is by its step response, which is, the response to a constant signal. A constant signal is represented by a step function, which is described as

$$u(t-a) = \begin{cases} 0 & t < a \\ 1 & t \geq a \end{cases} \tag{2-7}$$

The unit step function is the integral of the unit impulse function. Likewise, the step response is the integral of the impulse response. Quite often, the function used is equal to 1 for a finite period; that is, $u(t) = 1$ for $a \leq t \geq b$, for some $b>a$, and $u(t) = 0$ for $t>b$.

Sometimes in actual practice, rather than using the impulse function, a smooth input step function is generated. A smooth input step function is a function that does not switch values abruptly, as does the step function just defined, but rather that

changes values in a continuous manner. There are many examples in which a smooth step function simulates real-life situations more accurately than does the ideal step function. For example, the acceleration of a rocket at motor burnout does not vanish instantaneously, but tapers off to zero gradually.

As an example of using the step function, the smooth input step function shown in figure 2-1 can be used as a model to evaluate a typical general-purpose, recursive, second-order filter. (An nth order filter is a filter that can be designed with an nth order differential equation. The acceleration at rocket motor burnout discussed in the previous paragraph can be simulated using a second-order filter). The step response is shown in figure 2-2. A recursive filter must be given initial values, since each output value depends on the previous output value. The first three points in figure 2-2 represent the initial values used by the prospective filter.

2.4 Variance Reduction Factor

Let F be an arbitrary digital filter. Assume that the input and output of F are stationary random processes. A process is called stationary if its statistics such as mean and variance are not affected by a shift in the time origin. Theoretically, assume that F has been filtering the same process for an infinitely long time for the output to be stationary. Further, assume that each of the noise processes for the input and output data are uncorrelated with themselves. Let v(1) and v(2) be the variances of the input and output random processes. Then the variance reduction factor is defined as

$$R = \frac{v(2)}{v(1)} \tag{2-8}$$

The variance reduction factor must be interpreted with caution, because the assumptions underlying the estimation of variances are, in most cases, not realistic. The parameter R is not necessarily less than one, although R is generally less than one in a practical case.

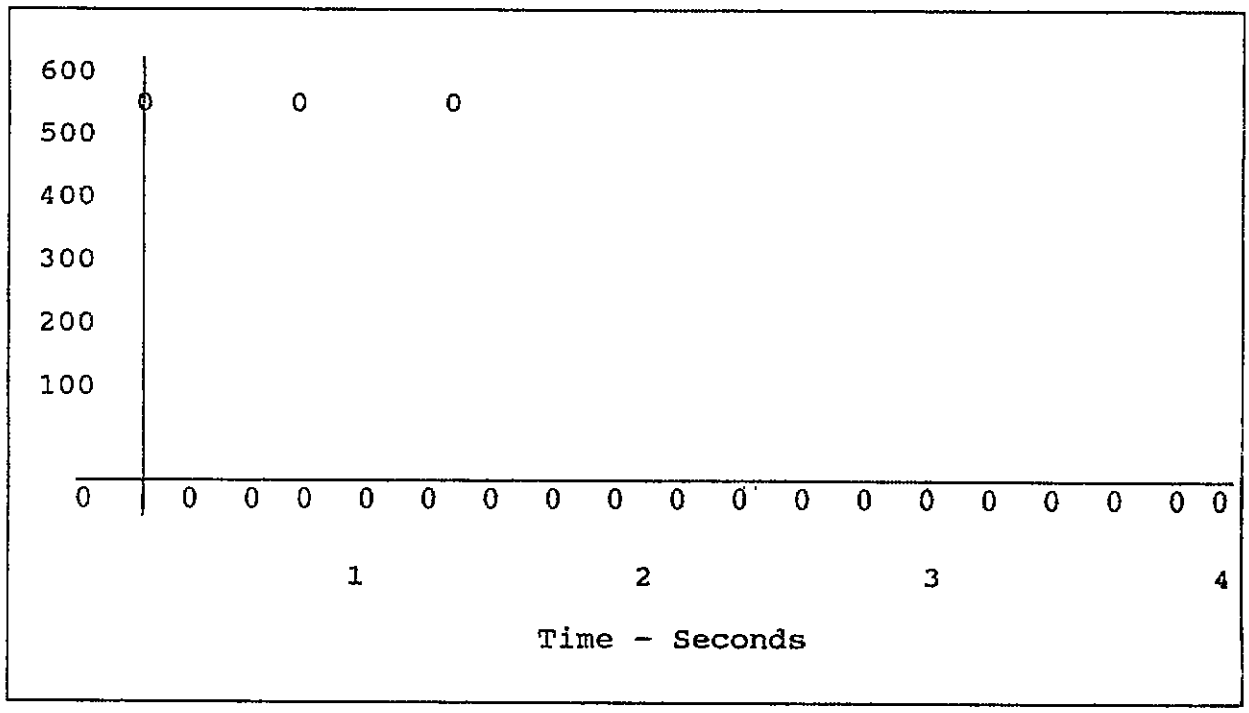

Figure 2-1. Smooth input data. Sampling period - 0.05 seconds.

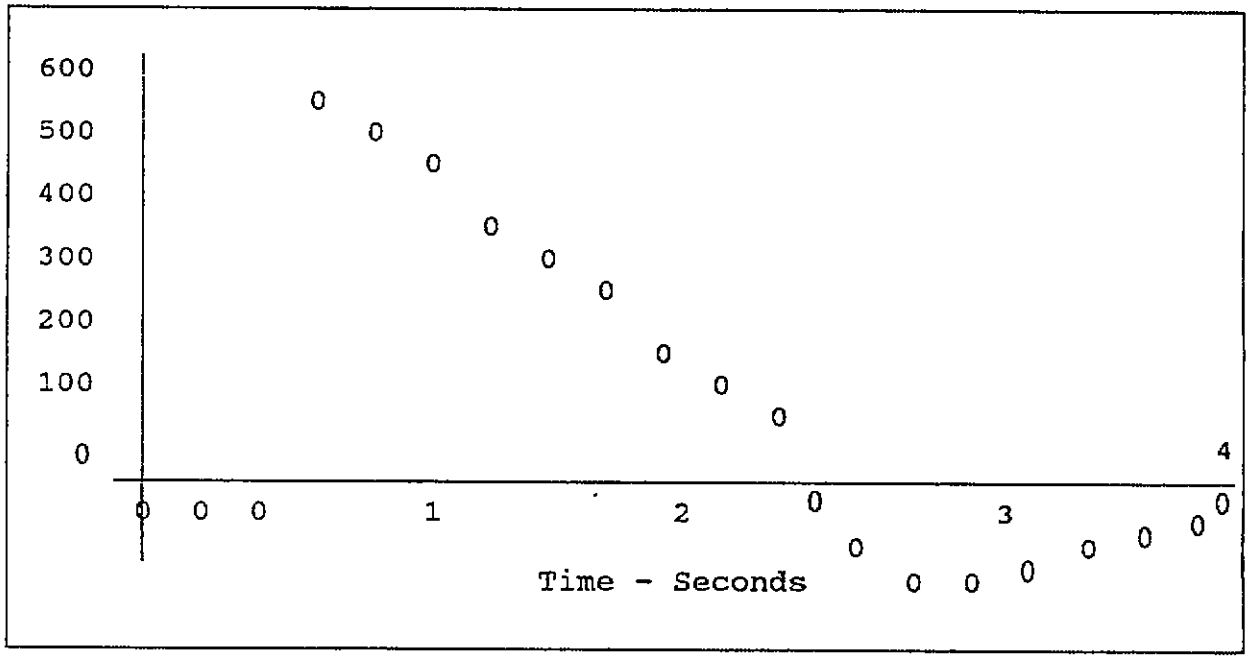

Figure 2-2. Real-time response. Point span - 20; sampling period - 0.05 seconds.

2.5 Simulation

Ordinarily, it is assumed that a signal process s(t) and a noise process n(t) defined for a discrete or continuous variable t are additive resulting in a process denoted by s(t) + n(t). A filter is used to suppress n(t) and extract s(t) with the least amount of distortion under a given set of operational criteria, for example, variance reduction factor, impulse response, and computation time. These factors usually vary from one process to another depending largely on the values of s(t) and n(t) even for a particular filter. It is therefore often desired to examine various response characteristics of a given filter applied to a specific signal plus noise process s(t) + n(t). The objective is to compare the filtered data, say $\hat{s}(t)$, obtained from s(t) + n(t) against the originally known signal process s(t). The act of perturbing s(t) with an additive noise process n(t) or filtering s(t) + n(t) or both is called the simulation of the signal plus noise process or of the filtering.

The simulation techniques are classified into two major categories: hardware simulations and numerical simulations. Simulated equipment tests and scale model experiments are typical of the hardware simulations. In the case of the numerical simulations, the entire system to be analyzed must first be represented by an appropriate mathematical model in the form of s(t) + n(t), in which the contribution of various perturbations is treated as the noise process n(t). The purpose here is to concentrate on numerical simulations.

The numerical simulations are divided into digital, analog, and hybrid. The digital simulations use digital computers as the primary simulation tools and are effective for discrete processes of the form s(i) + n(i), i=1,..., m. They are particularly suited for highly accurate simulation. Similarly, the analog simulations are made using analog computers or similar analog devices and are suited for the simulation of a process continuous in time. The hybrid simulations involve the mixture of both the digital and analog simulations incorporating digital-to-analog or analog-to-digital conversion processes or both. By the nature of their construction, neither the analog nor the hybrid computers can achieve a high degree of accuracy. Digital simulations have become more frequently used as the primary tools for analyzing systems performance and operational characteristics, even for those processes with continuous time variables.

Various digital filters may be applied to the output process of hardware simulations, digital simulations, or digitized values of analog/hybrid simulations. The performance of the filters is best evaluated under the conditions of the closed-loop simulation setup in figure 2-3. In a closed-loop simulation, the signal process s(t) is given a priori. The noise process n(t) is

generated using random noise generators or random number generators and is added to s(t), thus forming s(t) + n(t). The filter is applied to s(t) + n(t), yielding the filtered process $\hat{s}(t)$. The original signal process s(t) is fed into a comparator with appropriate delay to form the difference process $\hat{s}(t) - s(t)$, which can be evaluated for the performance of the filter.

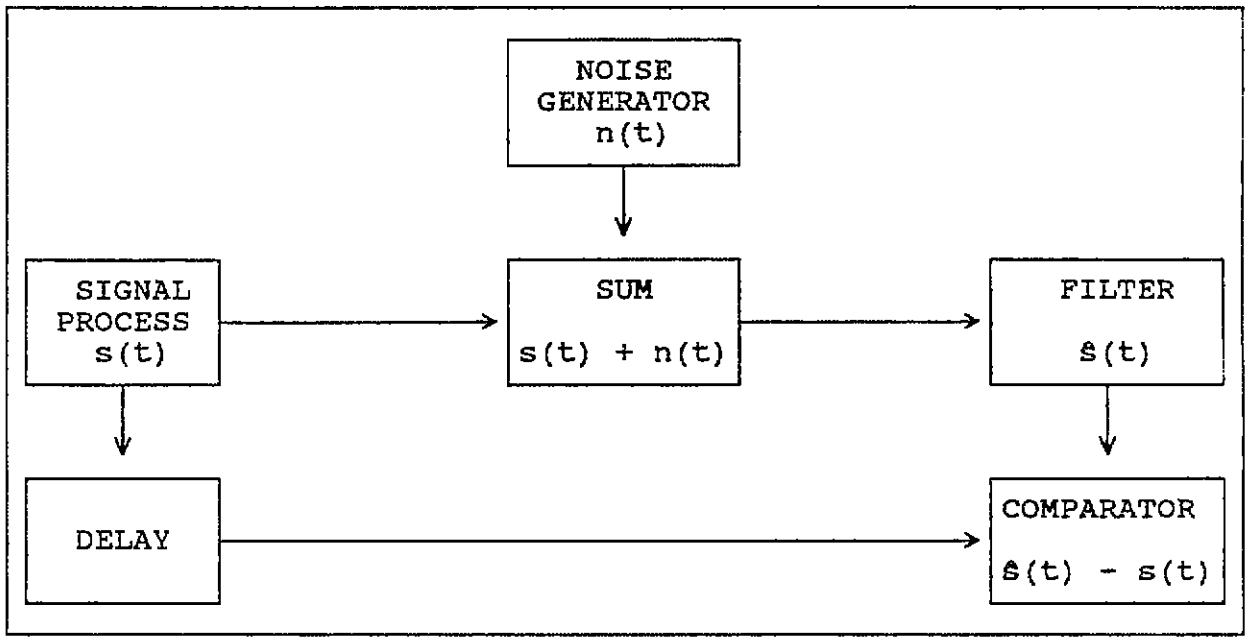

Figure 2-3. A closed-loop simulation logical diagram.

2.6 Monte Carlo Methods

An analogy of the random numbers used in digital simulations to the random outcomes of gambling devices such as roulette, dice, and cards has led to the use of the celebrated term "Monte Carlo" since the inception of digital simulations. The simulation of a random process by random numbers, followed by the calculation of statistical parameters of the end result of this process, is called a Monte Carlo technique. The random numbers in question can be generated by a computer. In the digital simulations of s(i) + n(i), i=1,..., m, the signal process s(i) is assumed to be known a priori and is available in the computer storage. The noise process n(i) is frequently assumed, unless facts to the contrary are known, to be independently and identically distributed random variables for all i=1,..., m with a given probability distribution function. The problem then reduces to the generation of random numbers with these qualifications.

2.6.1 Generation of Random Numbers From a General Distribution Function

Suppose {x(k): k = 1,..., }, are random numbers generated with a probability density function f(x), where f(x) can be either discrete or continuous in x. Basic concepts for the descrete and continuous cases are described in the following subparagraphs.

2.6.1.1 Definition of Discrete Case

If f(x) is discrete, it is then described as P{X = x}, where x is a discrete random variable (r.v.). There exists a probability value f(x(j)) for each random number x(j), j = 1, 2,..., ∞ such that

$$f(x(j)) \geq 0, \quad j = 1, 2, \ldots, \infty \tag{2-9}$$

and

$$\sum_{j=1}^{\infty} f(x(j)) = 1 \tag{2-10}$$

The cumulative probability F(x(n)) is defined to be

$$F(x(n)) = \sum_{i=1}^{n} f(x(i)) \tag{2-11}$$

It is the probability that the random variable X equals any one of the random numbers x(1), x(2),..., x(n). Notice that

$$\lim_{n \to \infty} F(x(n)) = 1.$$

2.6.1.2 Method of Random Number Generation

In the case of f(x) being continuous in x over the range of interest [a,b], f(x) is then described as having these three properties

$$f(x) \geq 0 \text{ for all } x \text{ in } [a,b] \tag{2-12}$$

$$\int_a^b f(x)dx = 1 \tag{2-13}$$

$$\int_c^d f(x)\,dx = P\{c \leq X \leq d\},$$

where X is a continuous r.v. and [c,d] is contained in $\subseteq$ [a,b]. The cumulative distribution function F(t) is defined to be

$$F(t) = \int_a^t f(x)\,dx, \quad a \leq t \leq b \tag{2-14}$$

Obviously, $F(a) = 0$, and $F(b) = 1$. (2-15)

2.6.1.3 Method of Random Number Generation

Let y(k), k=1,..., n, be random numbers uniformly distributed between 0 and 1. Random numbers x(k) can be generated from y(k) by finding a cumulative distribution function F mapping x to y, so that

$$y(k) = F(x(k)). \tag{2-16}$$

Pick F so that the new random numbers x(k) will be distributed in the manner desired. Now generate the new random numbers by using

$$x(k) = F^{-1}(y(k)). \tag{2-17}$$

In this manner, it is assured that the source y(k) of the new random numbers x(k) is random and uniformly distributed. The uniqueness of x(k) and the inverse mapping $F^{-1}y(k)$ can be seen immediately from figure 2-4. Equation (2-17) is applied often, since random number generators of many compilers produce numbers uniformly distributed between 0 and 1. The problem then narrows down to the generation of the random numbers y(k) from a uniform distribution over [0,1].

2.6.2 Random Numbers From a Uniform Distribution

Computers have mathematical routines which generate uniformly distributed random numbers. Various different earlier pseudo-random number generators have been examined in H. A. Meyer, Symposium on Monte Carlo Methods, John Wiley, New York, 1956.

One technique called the "congruence method" uses the following scheme. Given an initial random number x(k), $0 \leq x(k) \leq 2^n-1$, x(k+1) is computed recursively by

$$x(k+1) = (x(k)[1+2^a] \pm 1) \pmod{2^n} \tag{2-18}$$

where a = [n/2] ([·] is the greatest integer function). This technique is frequently used to generate random numbers in an n-bit binary machine.

2.6.3 Random Normal Deviates

Another kind of random number often used in digital simulation is that of the random normal deviates. A random normal deviate is a number in a set of random numbers that is normally distributed. Random normal deviates are used to simulate a random sample from a standard normal distribution. These random numbers can be generated using the mapping technique described in subparagraph 2.6.1 with the probability density function

$$f(x) = (2\pi)^{-1/2} \exp(-x^2/2). \qquad (2\text{-}19)$$

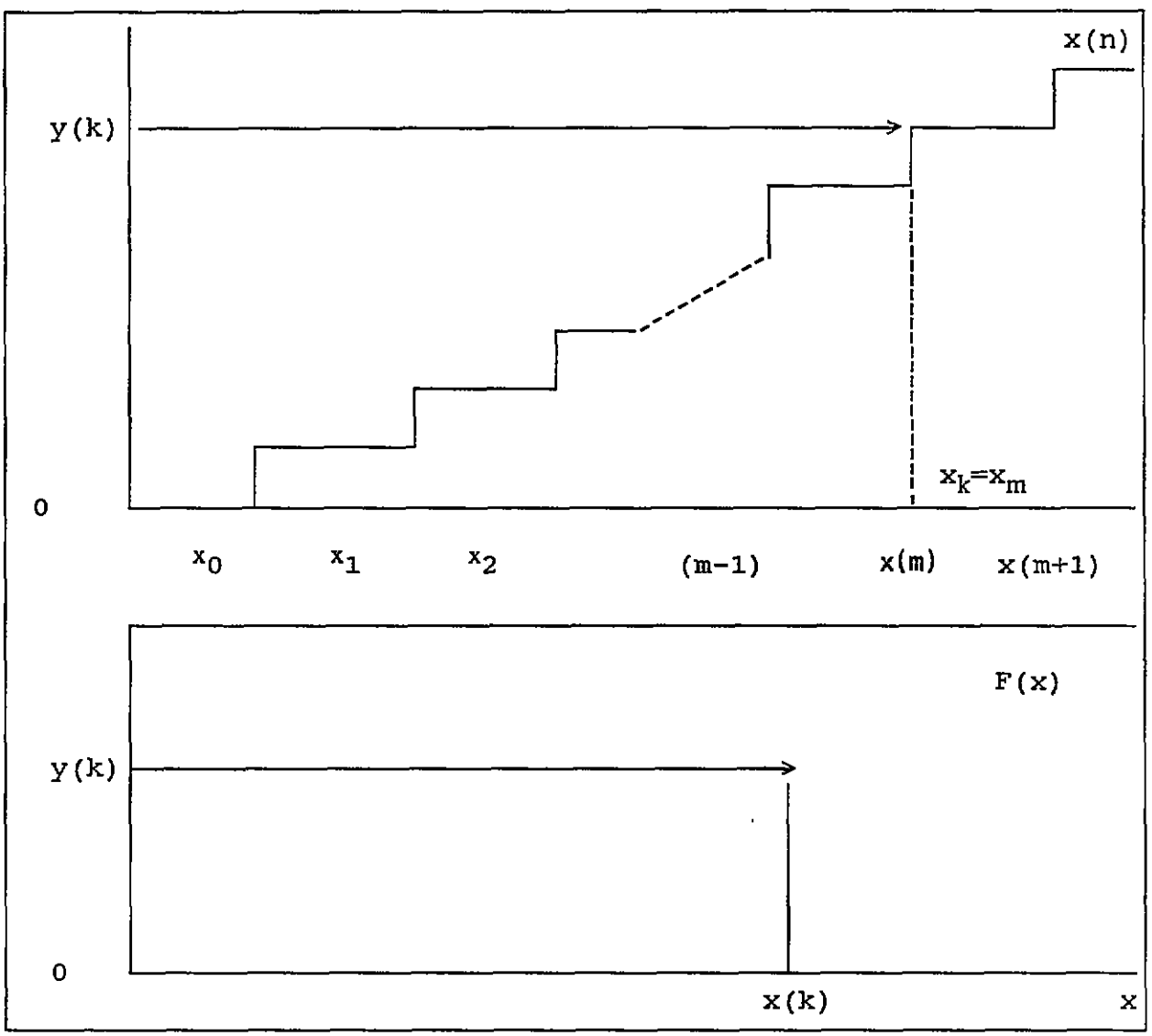

Figure 2-4. Graphic description of how the random number x(k) is determined. It is obtained with a cumulative distribution function F(x(n)) (in top graph) or F(x) (in bottom graph) and a random variable y(k) uniformly distributed over [0,1].

The resultant random numbers possess zero mean and unit variance. For the generation of random numbers $z(k)$, $k = 1,\ldots, N$, with a (nonstandard) normal distribution with mean m and variance v, use the transformation

$$z(k) = m + v(x(k)), \quad k = 1,\ldots, N. \tag{2-20}$$

2.7 Residuals

Some filter characteristics may be determined from using real test data by analyzing the residuals about the fit, that is, the differences between the filtered data values and the nonfiltered data values. A statistical analysis may be made for the autocorrelation and for the amount of variance or standard deviation. In addition, if the residual distribution includes a randomness about zero, this randomness is a good indication that the systematic error because of improper filtering is small. If these residuals do not display this randomness about zero, then the filter may be introducing bias or systematic error. The feature of randomness, however, in the residual analysis is only for the case when the noise is uncorrelated. Often the noise is correlated. Then the residual analysis becomes more complicated and must be undertaken with caution. Spectral analysis may be performed on the residuals as defined in RCC Document 153-71, Error Analysis and Methods For Estimating Errors in Position, Velocity, and Acceleration Data.

CHAPTER 3

FILTER ANALYSIS AND DESIGN IN THE FREQUENCY DOMAIN

3.1 Analysis of Filter Characteristics in the Time Domain

In the previous chapter, the analysis of filter characteristics in the time domain was considered. Also of interest are the characteristics of the filter in the frequency domain. To cite an instance, it may be necessary to know how the filter treats different frequencies which might be present in the data being filtered. Probably the most useful tool in any kind of analysis in the frequency domain is the Fourier transform as defined by

$$X(f) = \int_{-\infty}^{\infty} \exp(-2\pi ift)x(t)dt \qquad (3\text{-}1)$$

Here x(t) is some function of time, f is frequency, and $i = -1$. The function X(f), the Fourier transform of x(t), gives an indication of how different frequencies are distributed in the function x(t). To transfer from the frequency domain to the time domain, use the inverse Fourier transform, given by

$$x(t) = \int_{-\infty}^{\infty} \exp(i2\pi tf)X(f)df \qquad (3\text{-}2)$$

X(f) and x(t) are known as Fourier transform pairs.

3.2 Discrete Fourier Transform

Because the digital computer requires that input data be in sampled form, it is often more appropriate to use the discrete version of the Fourier transform given by

$$X(f) = t \sum_{j=-\infty}^{\infty} \exp(-2\pi ifj)x(j) \qquad (3\text{-}3)$$

where t is the time increment at which samples are taken. The variable t is sometimes referred to as the sampling period. Its reciprocal is sometimes referred to as the sampling frequency or sampling rate. The function X(f) is known as the discrete Fourier transform (DFT) of x(t). For a finite sequence

$$x(0), x(1),..., x(N-1) \qquad (3\text{-}4)$$

the discrete Fourier transform is given by the sequence

$$X(k) = \sum_{j=0}^{N-1} \exp(-2\pi ikj/N) \cdot t(x(j)) \quad k=0, 1, \ldots, N-1 \qquad (3\text{-}5)$$

The inverse discrete Fourier transform is given by

$$x(j) = (1/N) \sum_{j=0}^{N-1} \exp(-2\pi ikj/N) \cdot t(x(j)) \quad k=0, 1, \ldots, N-1 \qquad (3\text{-}6)$$

An efficient algorithm, developed by J. W. Tukey and J. W. Cooley published in 1965 and commonly used to calculate the discrete Fourier transform, is the Fast Fourier Transform (FFT). (A paper by I. J. Good a decade earlier describes a very similar algorithm.) Essentially, the FFT splits the sequence into two subsequences with each containing every other element of the original sequence, computes the discrete Fourier transforms (DFT) of those sequences, and rearranges the values of their DFTs back in their proper order to form the DFT of the original sequence. In this manner, the DFT can then be computed by using fewer complex operations (multiplications and additions).

Normally, the subsequences are divided again in the same manner as the original sequence to further reduce the number of operations required to compute the DFT of the original sequence. The subsequences are divided still further until there are only one-element sequences. When this subdividing has been completed, the ratio of the number of operations used for the FFT compared to the number of operations used to calculate the DFT directly is

$$\frac{\log_2 N}{N} \qquad (3\text{-}7)$$

where N is the order (length) of the original sequence.

It is often desirable that the original sequence have an order equal to a power of two, so it can be divided in the manner described earlier. In the event that the original sequence is not of such order, the sequence is often padded with zeroes, so that it can be completely reduced. The extra space used for padding the sequence with zeroes is more than compensated for by the savings afforded by the algorithm. Table 3-1 and figure 3-1 (obtained from references 47 and 53) demonstrate the difference between the number of multiplications used when calculating the DFT directly and that used when applying the FFT. See reference 52 for a brief and clear explanation of the Fast Fourier Transform.

TABLE 3-1. COMPARISON OF REQUIRED MULTIPLICATION OPERATIONS USING THE FFT AND THE DIRECT DFT.

N	N^2 (direct DFT)	$2N \log_2$ (FFT)
64	4,096	768
128	16,384	1,792
256	65,536	4,096
512	262,144	9,216
1,024	1,048,576	20,480

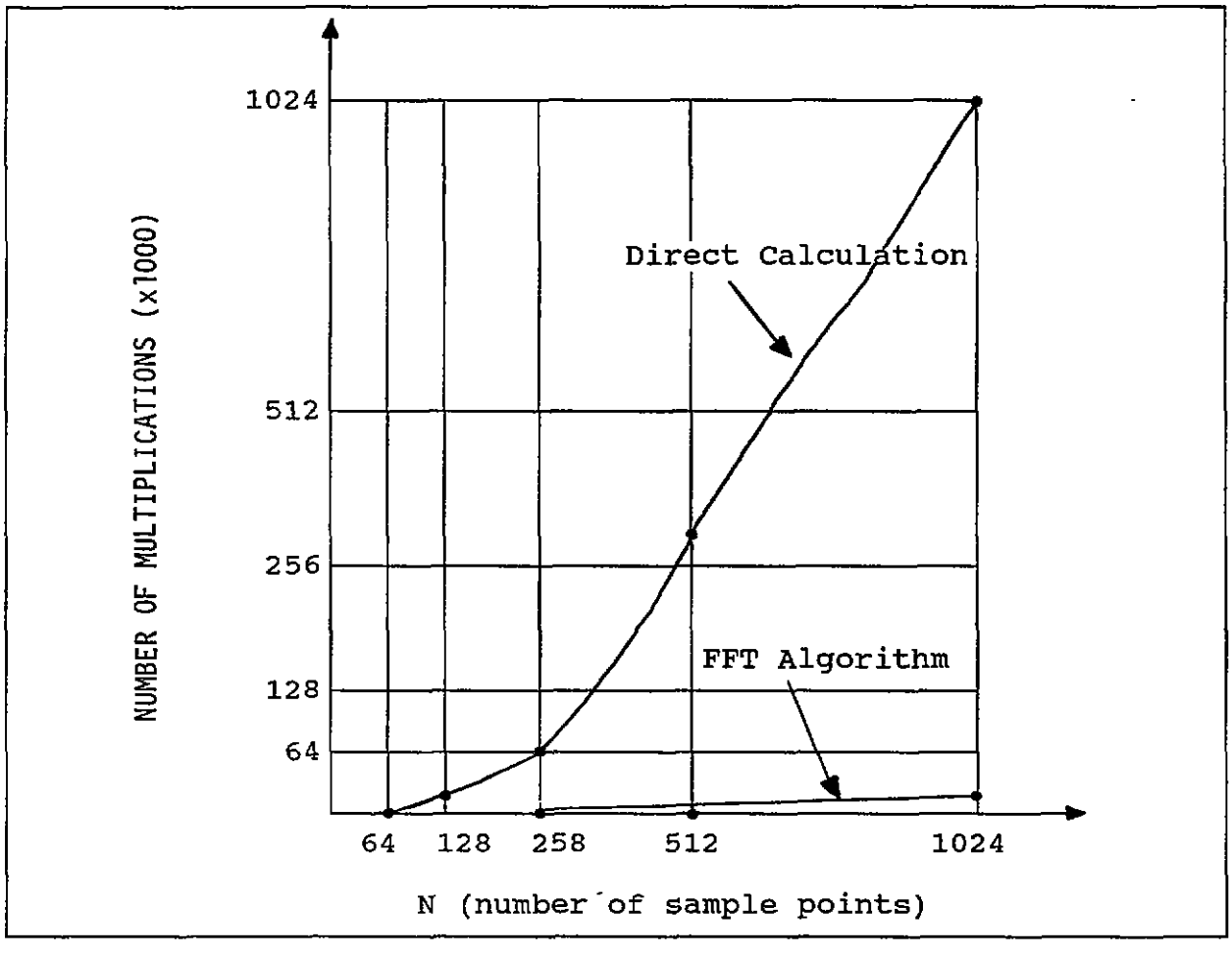

Figure 3-1. Comparison of multiplications required by direct calculation and FFT algorithm.

Another useful tool in the analysis of a filter frequency characteristics is the power spectral density function (psd) or power spectrum. This function's name came from a widely used term in the field of electrical engineering where it gives distribution of the signal power for each frequency. In this case, the power spectrum is measured in terms of watts per hertz. (In electrical engineering, the Fourier transform as a function of time is measured in hertz. Hertz is cycles per second.) The power spectral density function is the Fourier transform of the autocorrelation function, which is defined in chapter 2. In the statistical sense, it indicates how the variance is distributed as a function of frequency. For the discrete case, the power spectral density function of {x(j)} is given by the following sum

$$G_x(f) = \sum_{j=-\infty}^{\infty} \exp(-2\pi i f(j)) \cdot t(R(j)) \qquad (3\text{-}8)$$

where R(j), the autocorrelation function, is given by

$$R(j) = \sum_{i=-\infty}^{\infty} x(i)x(i-j) \qquad (3\text{-}9)$$

One of the best ways of analyzing a filter's frequency properties is to look at the frequency content of the output when the input is white noise. As mentioned in chapter 1, the power spectral density of white noise is a constant for all frequencies. Thus, by taking the power spectral density of the output of a filter whose input is white noise, it can be determined how the filter treats all the frequencies in the data. For example, it might be desired that the filter pass frequencies in a band $f_1 \leq f \leq f_2$ while rejecting all frequencies outside that band. Then, if the input is white noise, the ideal power spectral density of the output would be a translated rectangle as shown in the figure 3-2. In other words, if the psd of the output is that shown in the figure 3-2, then the appropriate frequencies (those less than f_1 or greater than f_2) were filtered out.

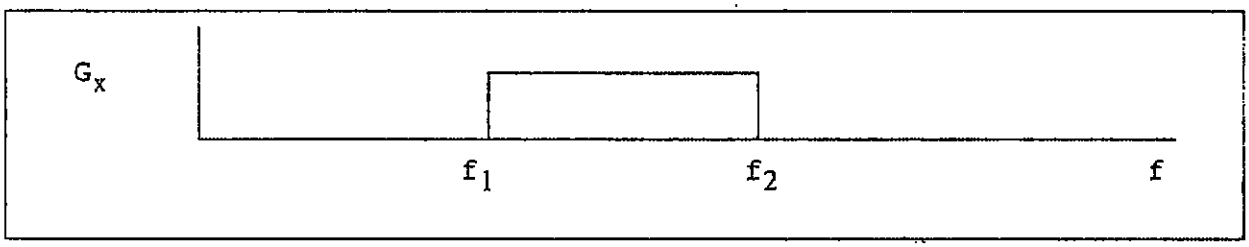

Figure 3-2. Power spectrum of output, where the input is white noise and where the filter passes only the frequencies between f_1 and f_2.

An indication of how good the filter is can be seen by comparing the power spectral density of the filter output, when the input is white noise, to this idealized power spectral density.

3.3 Transfer Function of a Filter

In chapter 1, it was noted that the output of a filter equals the input convolved with the filter weight function. A convolution for a time-invariant system takes the form

$$y(t) = \int_{-\infty}^{\infty} h(T)x(t-T)\ dT \qquad (3\text{-}10)$$

where h(T) is the weight function.

By taking the Fourier transform of the above equation, the equation becomes

$$Y(f) = H(f)X(f) \qquad (3\text{-}11)$$

The Fourier transform of the weight function, H(f), is known as the transfer function of the filter and represents the ratio of the filter output to the filter input in the frequency domain.

Repeating equation (1-18), the equation for a nonrecursive, time-invariant digital filter is

$$y(j) = \sum_{k} h(k)x(j-k) \qquad (3\text{-}12)$$

The transfer function H(f) for this filter is given by

$$H(f) = \sum_{k} h(k)\exp(-2\pi ifk)\cdot\Delta t \qquad (3\text{-}13)$$

Likewise, we repeat equation (1-19), the equation for a recursive time-invariant digital filter:

$$y(j) = \sum_{k=0}^{N} a(k)x(j-k) + \sum_{k=1}^{M} b(k)y(j-k) \qquad (3\text{-}14)$$

The transfer function H(f) for this filter is given by

$$H(f) = \frac{\sum_{k=0}^{N} a(k)\exp(-2\pi ifk)}{1 + \sum_{k=1}^{M} b(k)\exp(-2\pi ifk)} \qquad (3\text{-}15)$$

The modulus of the transfer function is called the gain of the filter and represents the amplification of the input.

Filters can be categorized according to the characteristics of their transfer functions. A filter is called an ideal low-pass filter if

$$|H(f)| = \begin{cases} 1 \text{ for } 0 \leq |f| < f_c \\ 0 \text{ for } f_c < |f| \end{cases} \qquad (3\text{-}16)$$

The plot of $|H(f)|$ is as shown in figure 3-3.

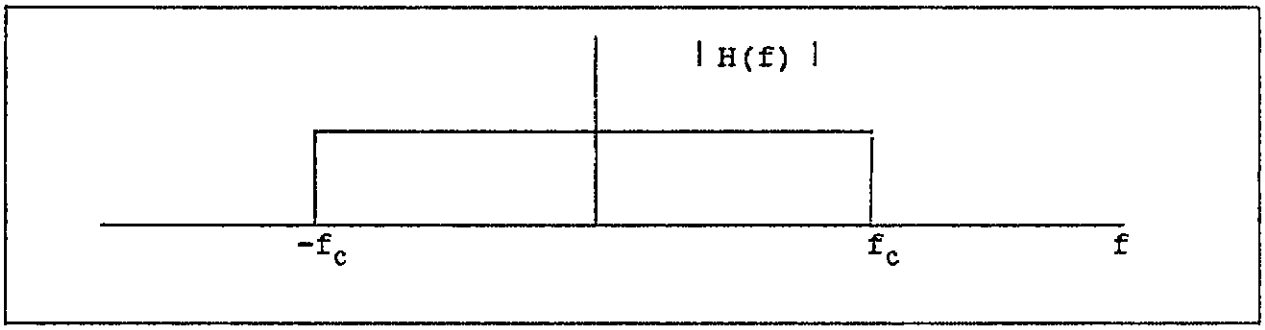

Figure 3-3. Ideal low-pass filter.

A filter is called an ideal high-pass filter if the transfer function is given by

$$|H(f)| = \begin{cases} 0 \text{ for } 0 \leq |f| \leq f_H \\ 1 \text{ for } f_H < |f| \end{cases} \qquad (3\text{-}17)$$

and the plot of $|H(f)|$ is as shown in figure 3-4.

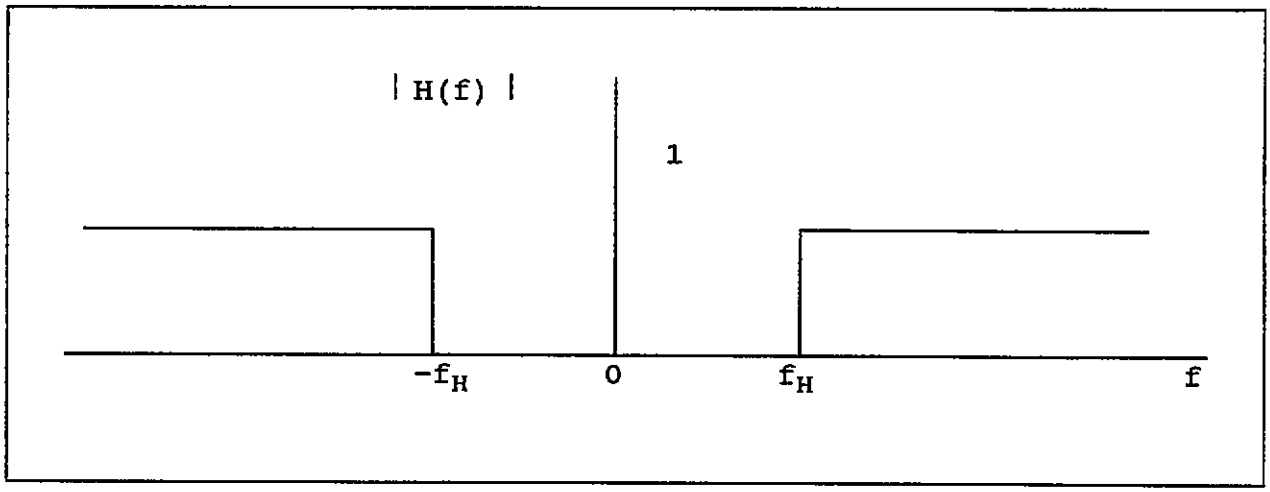

Figure 3-4. Ideal high-pass filter.

A filter is called an ideal band-pass filter if

$$|H(f)| = \begin{cases} 0 \text{ for } 0 < |f| < f_H \\ 1 \text{ for } f_H \leq |f| \leq f_L \\ 0 \text{ for } f_L < f \end{cases} \qquad (3\text{-}18)$$

and the plot is as shown in figure 3-5.

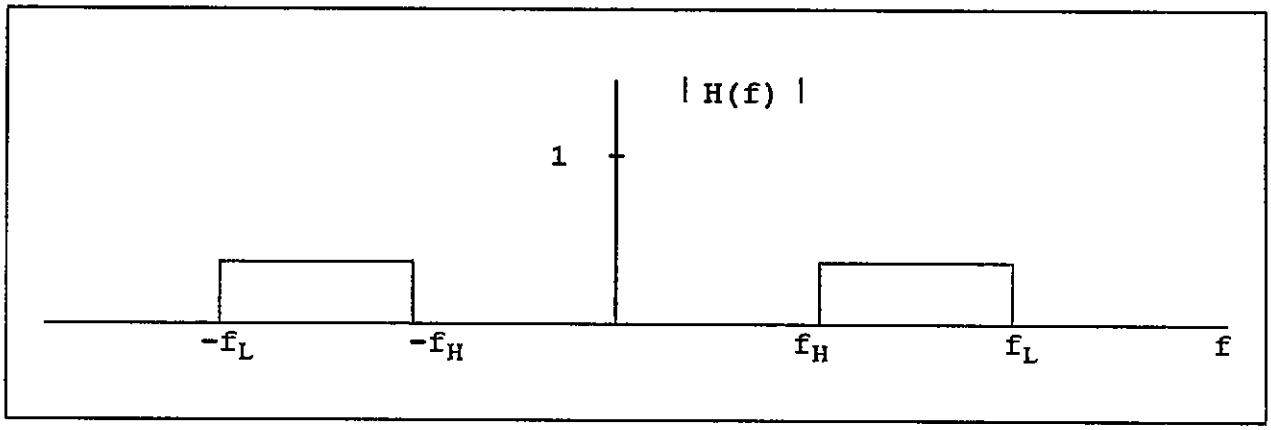

Figure 3-5. Ideal band-pass filter.

These ideal filters cannot be attained because there is, in practice, a finite number of samples. The impulse response in equation (3-10) is multiplied by a "boxcar" or window function which covers the time interval being sampled. Alternatively, the limits of integration can be changed to the appropriate, finite time limits. As stated in paragraph 1.5, the transform of the product of two functions is the convolution of the transforms of the two functions. Now the transform of the boxcar function is a sinc function. (By definition, sinc $x = \sin(\pi x)/\pi x$.) The resulting transfer function is the sinc function convolved with the transfer function of an ideal filter. The graph of the resulting gain is a boxcar function with ripples and not the ideal boxcar function illustrated in this section.

Methods for simulating the ideal low-pass filter are given in chapter 5. High-pass, band-pass, and band-rejection filters (discussed in chapter 4) are normally derived by first designing a low-pass filter. Methods for obtaining all of these filters are given in chapter 4.

3.4 Cutoff Frequency, Roll-Off, and Quality of Filter

In the previous section, a low-pass filter was defined in terms of a frequency denoted as f_c. This frequency f_c is called the cutoff frequency (see equation 3-16). The cutoff frequency is used as a criterion for designing digital filters in the frequency domain. It also gives a standard for analyzing the performance of filters in the frequency domain.

As explained in paragraph 3.3, ideal filters are simulated and not used directly. The simulating filters do not have gains that look like the boxcar functions drawn in paragraph 3.3. The gain of a simulating filter does not suddenly drop off from 1 to 0 the way the gain of an ideal filter does. Instead, it normally starts off (at f=0) with a value that is within a predetermined tolerance δ_1 of 1 and tapers off to within a predetermined tolerance δ_2 of 0.

The band of frequencies in which the gain is between $1-\delta_1$ and $1+\delta_1$ is called the passband. The greatest frequency value in the passband is the cutoff frequency for the passband. The band of frequencies in which the gain is $\leq\delta_2$ is called the stopband. The lowest frequency value for the stopband is the cutoff frequency for the stopband. The band in which the gain is $< 1-\delta_1$ and $> \delta_2$ is called the transition band. These items are illustrated in figure 3-6, which is obtained from reference 55.

One commonly used simulation to the ideal low-pass filter that is not designed on the basis of a transition band and two cutoff frequencies is the Butterworth filter. This filter and other simulations are given and explained in chapter 5.

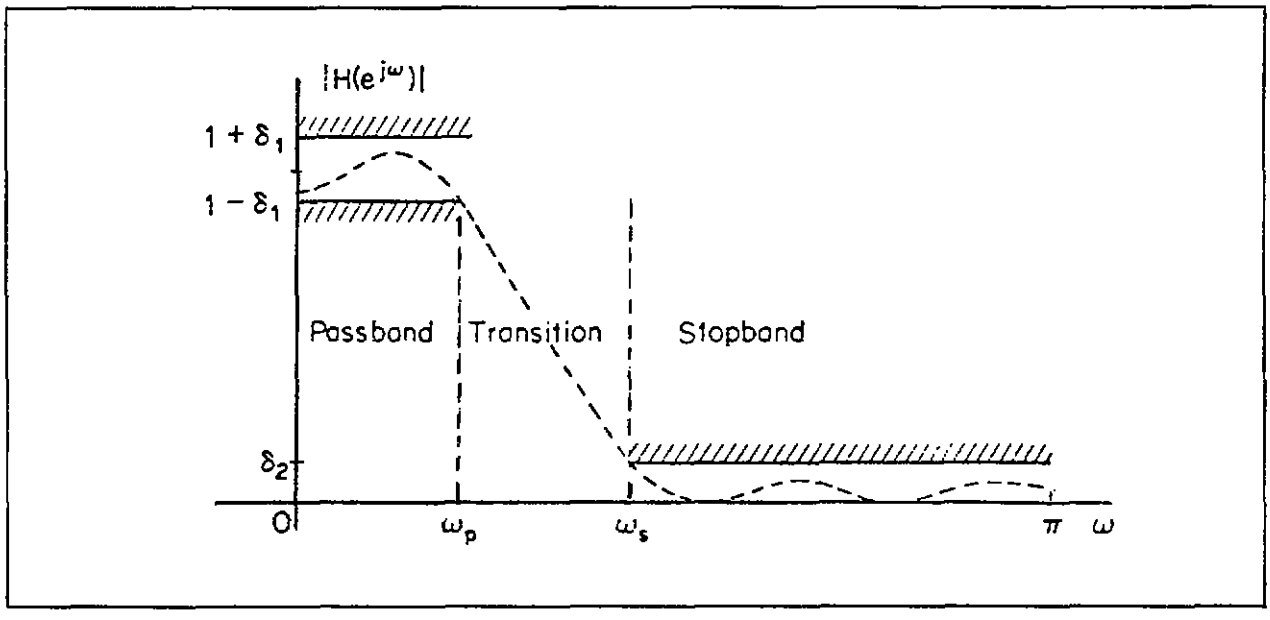

Figure 3-6. Tolerance limits for approximation of ideal low-pass filter.

Another important concept in the analysis is the roll-off. This quantity indicates to what extent the slope of the transfer function of a simulating (nonideal) filter (see paragraph 3.3) approaches the 90° slope of the transfer function of the ideal filter. Roll-off is related to the intuitive notion of "sharpness" of the cutoff. This quantity is defined as

$$p = \frac{-f}{|H|} \frac{d|H|}{df} \qquad (3\text{-}19)$$

A value for p can be obtained at any frequency desired. For purposes of comparison and analysis of filters, this quantity is often calculated at $|H| = 1/\sqrt{2}$. In figure 3-7, the plots of $|H(f)|$ for two filters, A and B, which approximate the ideal low-pass filter are given. Both filters have the same cutoff frequency, but A has a higher rolloff than B.

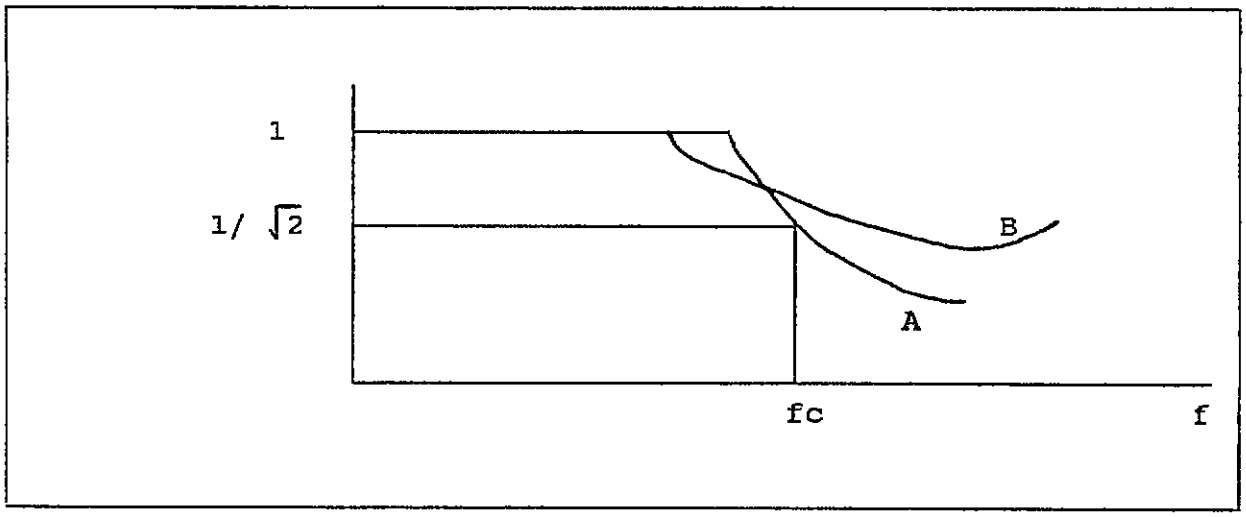

Figure 3-7. Comparison of rolloffs for filters A and B.

A third concept used in analyzing low-pass filters in the frequency domain is quality, which is defined as the extent to which the filter attenuates frequencies well above the cutoff frequency. (An attenuating function in the frequency domain is one that approaches zero as the frequency increases.) There is no standard mathematical definition for quality, so some subjective judgment must be used to determine the quality of a filter. One tool that can be used in making this judgment would be the power spectral density function as discussed in paragraph 3.2. For example, if two filters, A and B, yield the power spectral density functions shown in figure 3-8, then it might be concluded that filter A has the better quality.

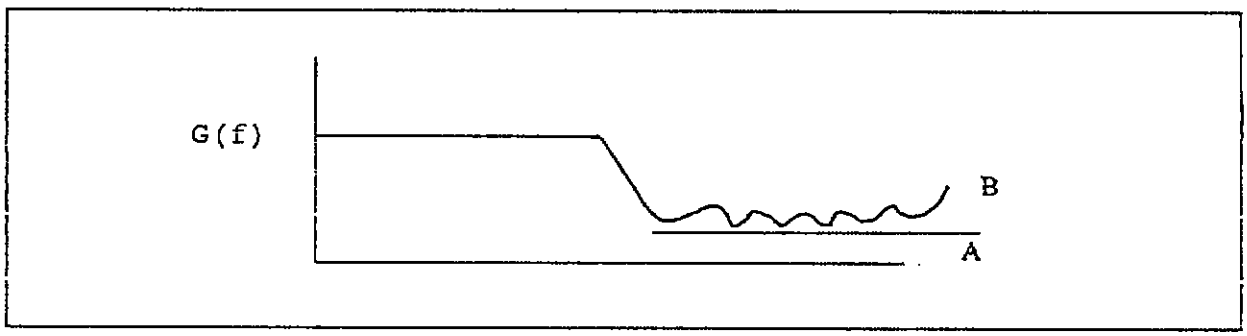

Figure 3-8. Comparison of the quality of filter A with the quality of filter B.

3.5 Phase Shift of a Filter

The transfer function H(f) of a filter is generally a complex number and can be expressed as

$$H(f) = |H(f)| \exp(i\theta(f)) \tag{3-20}$$

$\theta(f)$ is called the phase shift of the filter. Alternate names are phase response and phase angle. A time shift in the time domain corresponds to a phase shift in the frequency domain.

Equation (3-20) can be expressed as

$$H(f) = |H(f)| \cos(\theta(f)) + i\,H(f)\,\sin(\theta(f)) \tag{3-21}$$

or

$$H(f) = \mathrm{Re}(H(f)) + i \cdot \mathrm{Im}(H(f)) \tag{3-22}$$

where

$\mathrm{Re}(H(f)) = |H(f)| \cos(\theta(f)$ and $\mathrm{Im}(H(f)) = |H(f)| \sin(\theta(f))$. Re and Im denote real and imaginary parts of a complex number.

$$\text{Now} \quad \frac{\mathrm{Im}(H(f))}{\mathrm{Re}(H(f))} = \tan(\theta(f)) \tag{3-23}$$

The phase shift of the filter is then

$$\theta(f) = \arctan \frac{\mathrm{Im}(H(f))}{\mathrm{Re}(H(f))} \tag{3-24}$$

The output of the filter is in phase with the input only if $\mathrm{Im}(H(f)) = 0$. In most applications, this phase shift is important, and the analyst should be aware of its magnitude.

3.6 Construction of Filter Weights

Again consider a linear, nonrecursive digital filter whose defining equation is

$$y(k) = \sum_{j=o}^{M} h(j)x(k-j) \tag{3-25}$$

In this section, a general discussion of the way the $\{h(j)\}$ are determined will be given so that the filter will have a desired frequency response function. The frequency response is the same as the transfer function except that it is applied to a specific class of inputs in the form

$$x(n) = \exp(i2\pi fn). \tag{3-26}$$

$$x(n) = \exp(i2\pi fn). \tag{3-26}$$

One way of approaching this problem is to assume a desired frequency response function, $H(\omega)$, then take the inverse Fourier transform to get $h(t)$ as shown in equation (3-27). The variable ω is the angular frequency and is equal to $2\pi f$.

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(i\omega t)H(\omega)\, d\omega \tag{3-27}$$

The weights for the discrete filter are then determined by evaluating this continuous weight function at specific instants in time,

$$h_n = h(t_n)\Delta t. \tag{3-28}$$

There are problems with constructing these weights if the ideal frequency response is used. For example, the ideal frequency response for the low-pass filter is the unit rectangle as given in figure 3-3. The inverse Fourier transform must be truncated in the applied case, so only a finite number of weights are used in equation (3-25). The truncated Fourier transform of the unit rectangle is very "bumpy" (see paragraph 3.3, page 3-7) because of the discontinuity at f_c and is not useful in most applications.

Approaches to avoiding this dilemma are given in references 7, 11, and 13. In reference 7, Gennery eliminates the dilema by using various modifications of the Gaussian distribution function for the ideal response function. In reference 11, Ormsby replaces the ideal frequency response function with a function having a finite slope. In reference 13, Stirton multiplies the $h(t)$ for the ideal filter by certain "apodizing" functions which eliminate the bumps in the frequency response function.

In certain applications, the z-transform rather than the Fourier transform is used for designing filters in the frequency domain. The z in the z-transform is often used as a replacement for the exponential function in the discrete Fourier transform (DFT). It is used as such for computational ease and to facilitate transfer function analysis. For example, it is easier to determine stability of a filter by looking at the z-transform rather than by looking at the DFT. Determining filter stability in this manner will be discussed again in paragraph 3.8 and in chapter 5.

The z-transform approach is used when the transfer function of an analog filter is known, and when construction of a digital filter having that same transfer function is desired. The z-transform of an analog function $f(t)$ is defined as

$$\mathcal{Z}(f(t)) = \sum_{n=0}^{\infty} f(nT)z^{-i} \qquad (3\text{-}29)$$

It can be shown that the z-transform of the weight function is the ratio of two polynomials in the form

$$H(z) = \frac{\sum_{n=0}^{N} a(n)z^{-i}}{1 + \sum_{n=1}^{M} b(n)z^{-i}} \qquad (3\text{-}30)$$

if the filtered output {y(k)} is given by the recursive relation

$$y(k) = \sum_{n=0}^{N} a(n)x(k-n) + \sum_{n=1}^{M} b(n)y(k-n) \qquad (3\text{-}31)$$

It can also be shown that the z-transform of the weight function is in the form

$$H(z) = \sum_{n=0}^{N} a(n)z^{-i} \qquad (3\text{-}32)$$

if the filtered output {y(k)} is given by the nonrecursive relation

$$y(k) = \sum_{n=0}^{N} a(n)x(k-n) \qquad (3\text{-}33)$$

Other ways in which H(z) can be expressed are discussed in chapter 5.

3.7 Aliasing

In performing with digital filters, samples are taken from an analog sinusoidal waveform, and the waveform is reconstructed from the samples. The samples are typically taken at equal-spaced time intervals. The problem arises about how short these sampling intervals should be so that the original analog waveform can be exactly reconstructed. If the sampling intervals are too large, that is, if the samples are too infrequent, the original analog waveform may be reconstructed to have a lower frequency than it actually has. This misrepresentation of a frequency by a lower frequency is known as aliasing.

The appearance of wheel spokes turning slower than they actually are is an example of aliasing. The alternate term "foldover" can be expressed by the next example. Suppose an observer is taking visual samples at a constant time rate of the fast-turning wheel spokes. As the wheel's rotational speed increases, the wheel spokes seem to be rotating faster up to a certain point. Then the wheel spokes appear to slow down until they seemingly rotate faster and faster in the backward direction up to a certain point. The spokes then seemingly slow down until it looks as though they are going in a forward direction. If the observed rotational motion is placed on a plot of observed frequency versus actual frequency, the graph "folds over" at a frequency value known as the Nyquist or folding frequency (see figure 3-9). The F_o axis represents the observed frequency while the f_a axis represents the actual frequency, and the value f_N is the Nyquist frequency.

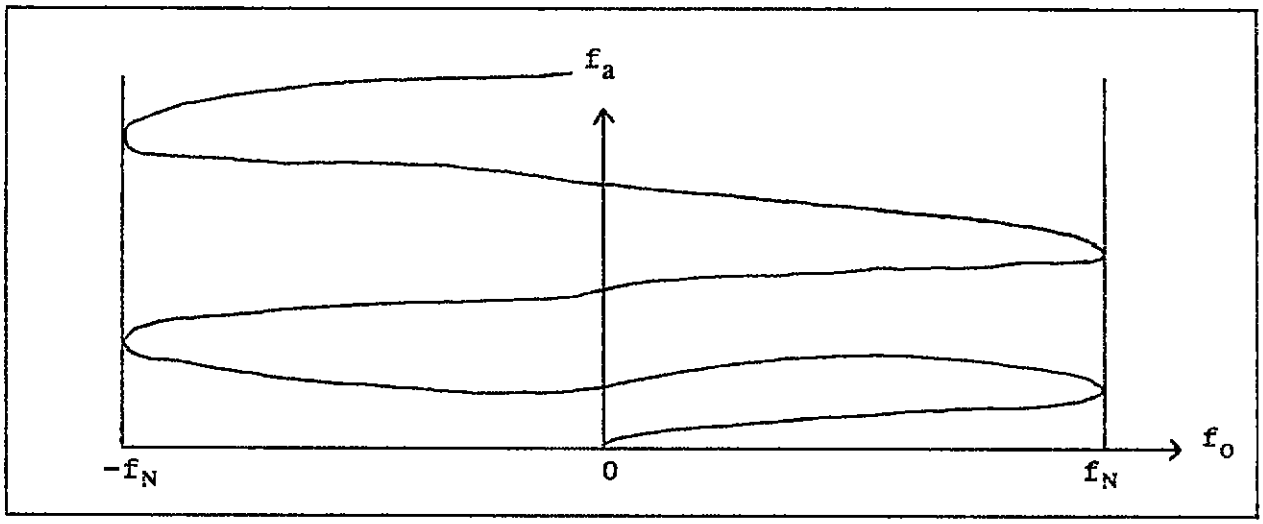

Figure 3-9. Foldover or aliasing resulting from observing a wheel's increasing rotational speed.

3.7.1 Aliasing in the Time Domain

An illustration of aliasing in the time domain is given in figure 3-10. The original analog waveform is the thinner sinusoid and is sampled at the points shown. With these sampled points, the original waveform is reconstructed so that the result is the thicker line which, of course, has a lower frequency than the original waveform. In producing new waveform, the original one was misreconstructed because the samples were taken too infrequently (see figure 3-10).

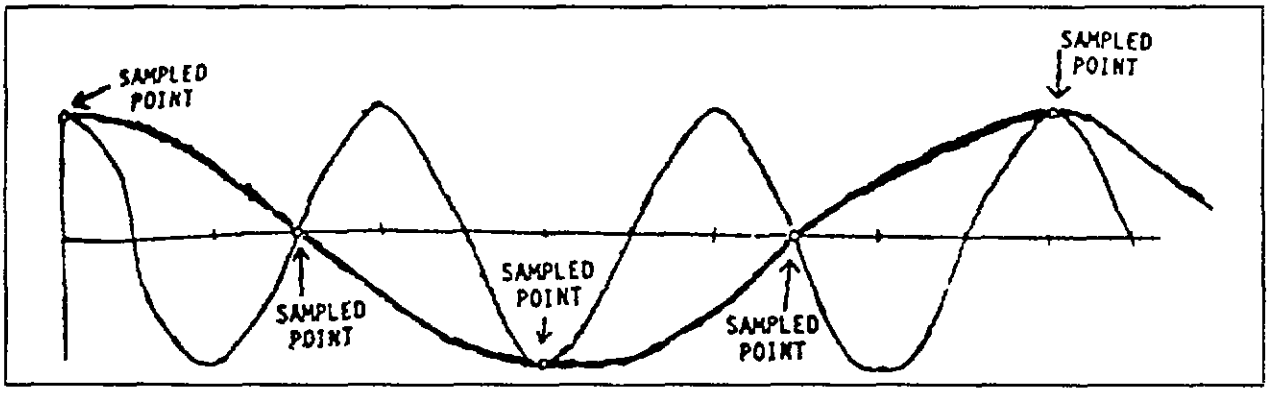

Figure 3-10. Aliasing in the time domain.

More often than not, the original waveform represents the sum of many sine and cosine functions. Figure 3-11, obtained from reference 52, illustrates this notion. The bottom sinusoid is the sum of the top three sinusoids. The frequency of each of the top three sinusoids is called a "frequency component" of the bottom sinusoid.

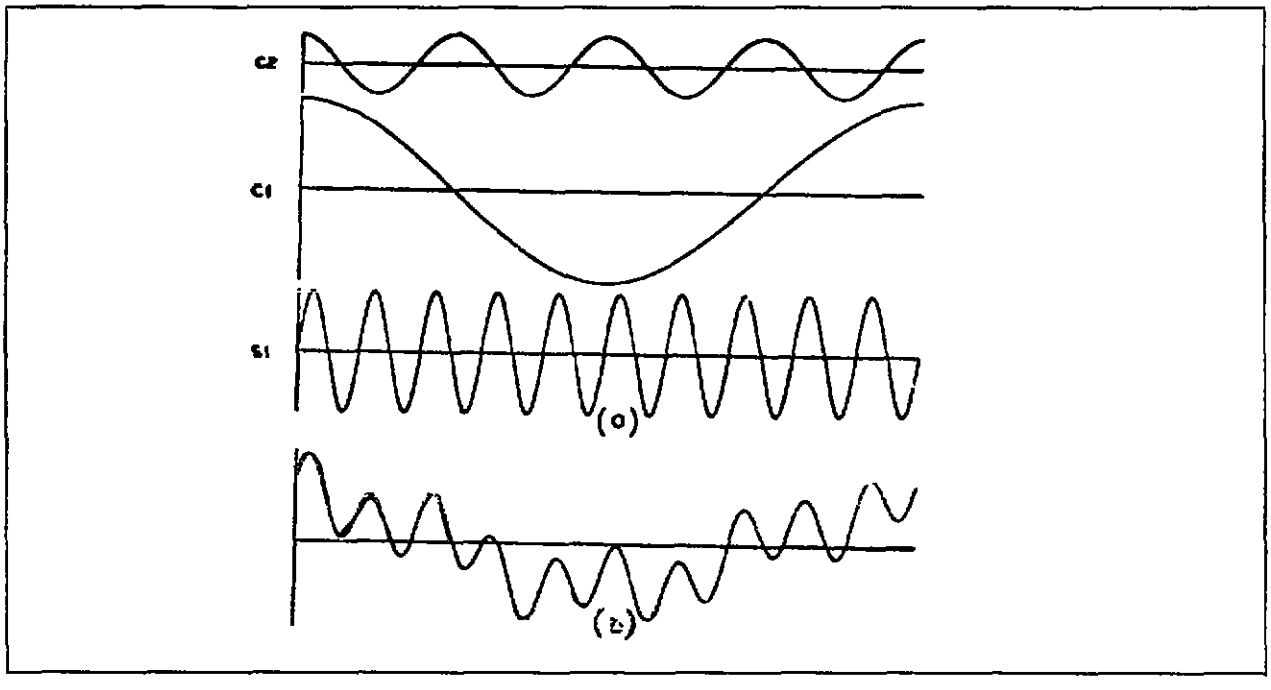

Figure 3-11. Bottom sinusoid shown as the sum of three frequency components, C2, C1, and S1.

3.7.2 Aliasing in the Frequency Domain

The discrete Fourier transform of a signal can be expressed in terms of the analog Fourier transform by

$$X(e^{i\omega T}) = (1/T) \sum_{m=-\infty}^{\infty} X_A(\omega + (2\pi/T)m) \qquad (3\text{-}34)$$

where $X_A()$ (in the right-hand side) is the analog Fourier transform, and $X()$ (in the left-hand side) is the discrete Fourier transform. This equation is derived in reference 55.

The frequency interval of each term in the summation sign of equation (3-34) will not intersect with its adjacent interval if it is at most $2\pi/T$ wide. Figure 3-12, obtained from Reference 54, point to this fact. Figure 3-12 illustrates the analog Fourier transform of one of the terms, which covers a frequency bandwidth of $2\pi/T$ and illustrates the infinite sum of such terms. As can be seen, $2\pi/T$ is the widest that these terms can be without intersecting each other.

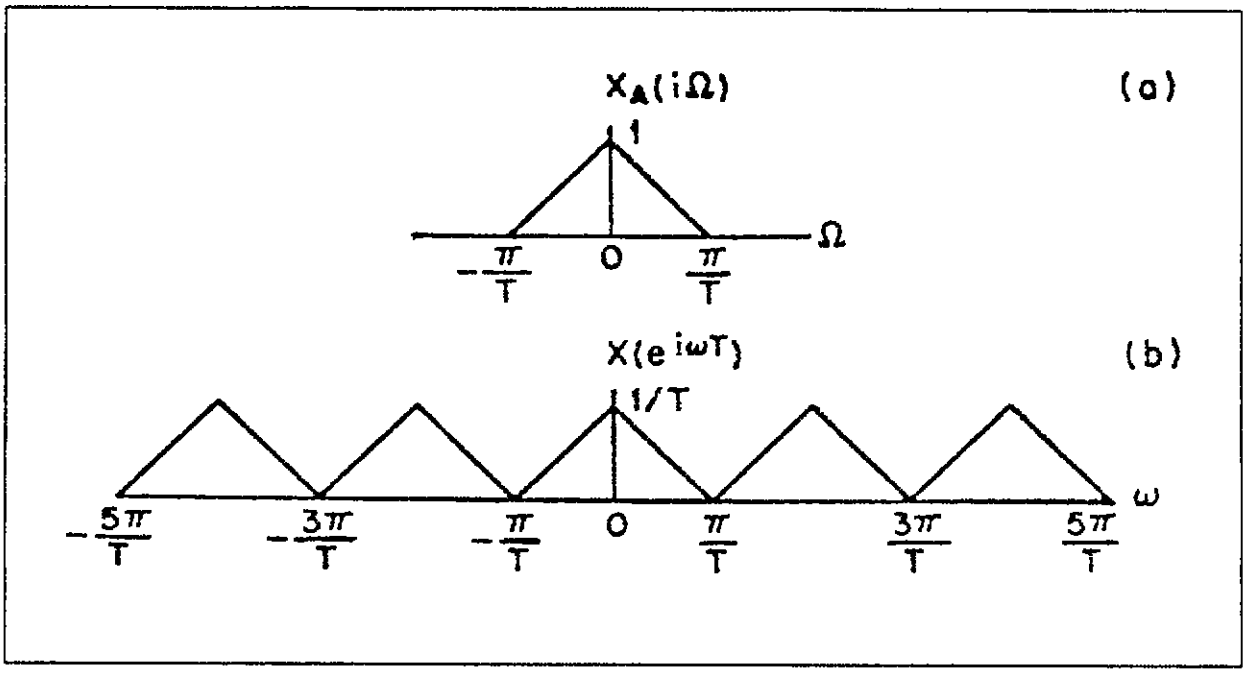

Figure 3-12. Sampling relations for analog and digital systems for properly sampled inputs.

Figure 3-13, obtained from reference 54, gives an example where samples are taken too infrequently, in particular, where they are taken at a rate of $w/3\pi$, which is less than $w/2\pi$. In other words, the frequency bandwidth for each term, which is $3\pi/T$, is greater than $2\pi/T$.) Figure 3-13a shows the graph of one of the terms in the summation of equation (3-34), when substituting $3\pi/T$ for w. Figure 3-13b explains how the adjacent terms overlap each other, and figure 3-12c displays the resulting graph of the Fourier transform. It is easily seen how the frequency $3\pi/2T$ can be mistaken for the frequency $\pi/2T$. This phenomenon, where in effect a frequency component takes on the identity of a lower frequency, is aliasing or foldover.

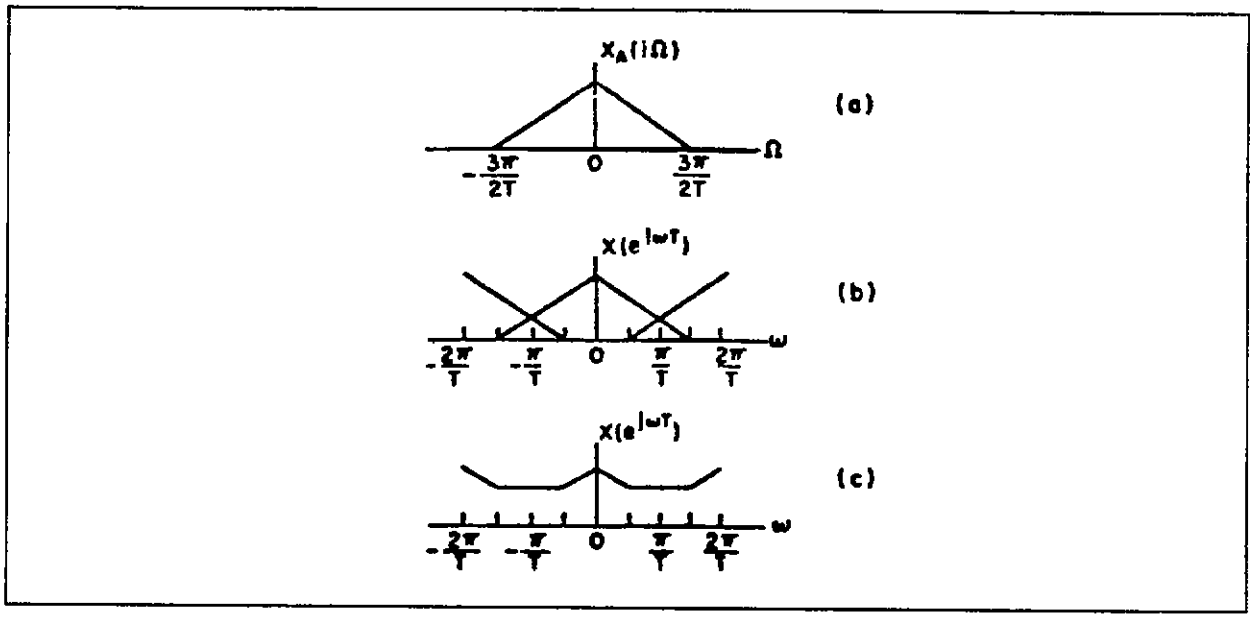

Figure 3-13. The effects of undersampling on the digital frequency response.

3.7.3 Sampling Theorem and the Nyquist Frequency

The Sampling Theorem offers a way to avoid aliasing. It states that if the Fourier transform of a signal is bandlimited, then the original signal can be exactly reconstructed if samples are taken at a frequency of at least twice that of the highest frequency component in the signal. In mathematical terms, the Sampling Theorem states if $F(\omega)$, the Fourier transform of a periodic time function f(t), is bandlimited so that

$$F(\omega) = 0 \text{ for } |\omega| \geq \omega_c \quad (\omega_c = 2\pi f_c) \tag{3-35}$$

then f(t) can be uniquely determined if it is sampled at intervals no greater than $1/2f_c = \pi/\omega_c$, that is, at a sampling rate of no less than $2f_c = \omega_c/\pi$.

Figures 3-11 and 3-12 imply the validity of this theorem. From these figures, it can be seen that the frequency ω_c should be at most π/T to avoid foldover. In other words, the sampling rate $1/T$ should be no less than ω_c/π.

For samples f_n which equals $f(t_n)$, where $t_n \leq n/2f_c = n\pi/\omega_c$, then the original analog signal f(t) can be reproduced by

$$f(t) = \sum_{n=-\infty}^{\infty} f_n \frac{\sin(\omega_c t - n\pi)}{\omega_c t - n\pi} \qquad (3\text{-}36)$$

$2f_c$ is known as the Nyquist or folding frequency. (This theorem is stated and elegantly proven in subparagraph 9.1.1 of reference 47.)

3.8 Stability of Filter

A digital filter is stable if every bounded input sequence yields a bounded output sequence through this filter. A sequence is bounded if all its terms are less than a specific positive integer. It can be shown that a necessary and sufficient condition for stability is that the impulse response $|h(k)|$ be such that

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \qquad (3\text{-}37)$$

From this condition, it can be shown that another indication of filter stability is that all the poles of the transfer function be inside the unit circle in the (complex) z plane. If poles are on the unit circle, then the filter may or may not be stable. If a pole is outside the unit circle, the filter is not stable. See chapter 5 for the relation between pole location and filter stability.

CHAPTER 4

CATEGORIES OF DIGITAL FILTERS

In Chapter 1, some of the fundamental principles of digital filters were discussed. As stated earlier, the basic reason for using a digital filter is to separate or suppress errors and to pass signals without significant distortion. As discussed in Chapter 3, the frequency response of the filter should be such that those frequencies consisting mainly of the desired signal should be passed through the filter, and those frequencies consisting mainly of noise (error) should be rejected. If the complete statistical characteristics of the signal and noise are known, then using the principles discussed in chapter 3, a filter can be constructed to separate signal from noise in an optimum manner.

4.1 Low-Pass Digital Filters

The data from missile trajectory work usually contains a signal which consists mainly of large low-frequency components. The noise error is usually assumed distributed throughout the frequency spectrum with the desired signal being much greater in amplitude than the noise at low frequencies but smaller than the amplitude of the noise at the high frequencies. In this case, a low-pass filter is designed to pass the desired signal.

A low-pass filter has a frequency response of exact unity at zero frequency, approximate unity at low frequencies, and approximately zero at higher frequencies (see figure 4-1). The low-pass filter is simulated by the ideal low-pass filter defined in paragraph 3.3). The frequency at which the transition occurs between the high and low frequencies is called the cutoff frequency (f_c) of the filter. Because a low-pass filter removes the high frequency fluctuations from the data, it is often referred to as smoothing the data.

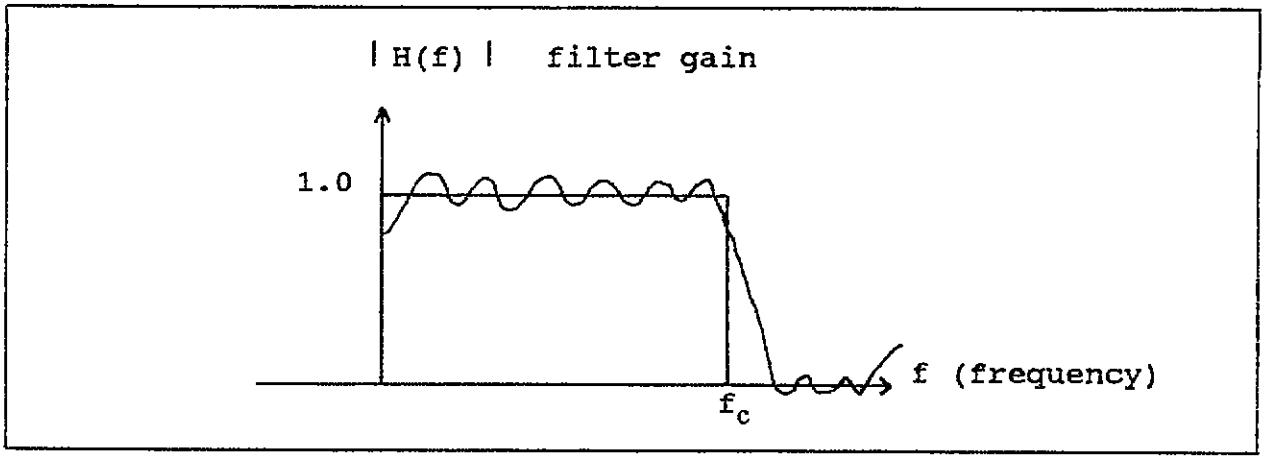

Figure 4-1. Typical frequency response of low-pass filter (represented by the curved line).

4.2 High-Pass Digital Filters

The category of low-pass filters discussed in the previous section can be transformed into filters which perform other tasks. In analysis work, one of the most commonly used is the high-pass filter. As the name implies, a high-pass filter is designed to pass high frequencies and reject the low frequencies. A low-pass filter can be transformed into a high-pass filter in several conceptually equivalent ways. One of the most common way is to subtract its frequency response from unity at all frequencies. That is, if $H_L(f)$ is the frequency response of a low-pass filter, then

$$H_H(f) = 1 - H_L(f) \qquad (4\text{-}1)$$

defines the frequency response of a high-pass filter.

Since

$$H_L(0) = 1 \qquad (4\text{-}2)$$

for a low-pass filter, then it follows that

$$H_H(0) = 0 \qquad (4\text{-}3)$$

Equation (4-1) shows that output from a high-pass filter is the same as the difference between the unfiltered data, that is, output from an "all-pass" filter and the output from a low-pass filter. Clearly, since a filter whose response is unity over all frequency bands, an all-pass filter passes the data without

change; that is, it does not filter the data at all. Therefore, the effects of a high-pass filter can be duplicated by filtering the raw data with a low-pass filter and then subtracting the results from the raw data as in

$$Y = H_H X = (1-H_L)X = X - H_L X \qquad (4-4)$$

X is the raw data (in the frequency domain). $H_L X$ is the low-pass filtered data. This technique is often used in analysis of high-frequency noise on missile trajectory data; however, the procedure of subtracting the low-pass or smoothed data from the raw data can be cumbersome and time consuming. A generally faster approach is to construct the weights of the high-pass filter and apply them directly to the raw data.

The weights for this high-pass filter can be obtained by taking the difference between the weights for the two filters given by the right hand side of equation (4-1). Remember from paragraph 2.3 that these weights amount to the unit impulse responses of their respective filters. For an all-pass filter, the response to a unit impulse is a unit impulse, because the input data is left unchanged by this filter. Consequently, all the weights of an all-pass filter are unit impulse responses. The weights for the low-pass filter are then subtracted from the weights of the all-pass filter to obtain the weights for the high-pass filter. Letting $W_{L,k}$ represent the weights for a low-pass filter and $W_{H,k}$ represent the weights for the corresponding high-pass filter, then

$$W_{H,k} = \delta_k - W_{L,k} \qquad (4-5)$$

where

$$\delta_k = \begin{cases} 1 \text{ if } k = 0 \\ 0 \text{ if } k \neq 0. \end{cases} \qquad (4-6)$$

4.3 Band Pass Filters

The band-pass filter passes a set of adjacent frequencies while rejecting the frequencies above and below the set of frequencies. A band-pass filter is said to be narrow if the frequency band contains very few frequencies, and wide if it contains many frequencies. If the desired frequency bandwidth is large compared to the center frequency of the band, a wide band-pass filter would be desirable to use for filtering the band. If the bandwidth is small compared to the center frequency, it would be better to filter it with a narrow band-pass filter.

The wide band-pass filter is constructed by combining a high-pass filter with a low-pass filter. The high-pass filter, used for this construction, has a cutoff frequency f_c in the area of the spectrum where the low-pass frequency response is approximately unity. The low-pass filter that is used has a cutoff frequency f_{c_H} where the frequency response of the high-pass filter is approximately unity. There is a resulting band of frequencies where the low-pass and high-pass filters are both unity, so that $f_{c_L} > f_{c_H}$. The frequency response of the resulting filter would have a low end of f_{c_H}, a high end of f_{c_L} and a center frequency of

$$f_o = \frac{f_{c_L} + f_{c_H}}{2} \quad \text{(see figure 4-2).} \qquad (4\text{-}7)$$

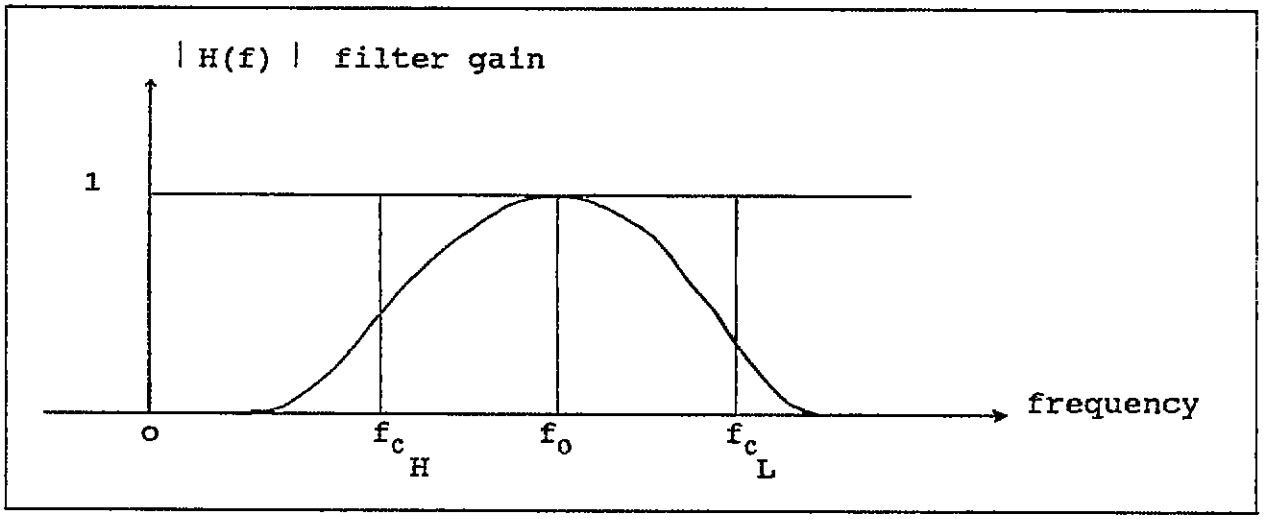

Figure 4-2. Frequency response of wide band pass filter.

To perform the band-pass filtering, the high-pass and low-pass filters can be applied separately to the data in series, which is usually time consuming. It is more desirable to combine the two filters into a single band-pass filter by convolving the weights of the two filters with the input data in the time domain. The same results can be obtained by multiplying the frequency response of each filter with the Fourier transform of the input data in the frequency domain and then taking the inverse Fourier Transform to get the band-pass filtered data in

the time domain. The methods mentioned here are essentially the same as the methods described in paragraph 4.2 for converting a low-pass filter into a high-pass filter.

If the frequency response curves of the high-pass and low-pass filters are sharp and closely approximate the ideal frequency response of a step function, the techniques discussed previously will work for a narrow band-pass filter. If this is not the case and the roll-off is not steep, a narrow band-pass filter can be constructed using the following technique. The frequency response of a low-pass filter is translated so that the center frequency of the low-pass filter, f=0, is moved to the desired center frequency of the passband with its own mirror image reflected about the zero frequency, thus its mirror image is at the corresponding negative frequencies (see figure 4-3).

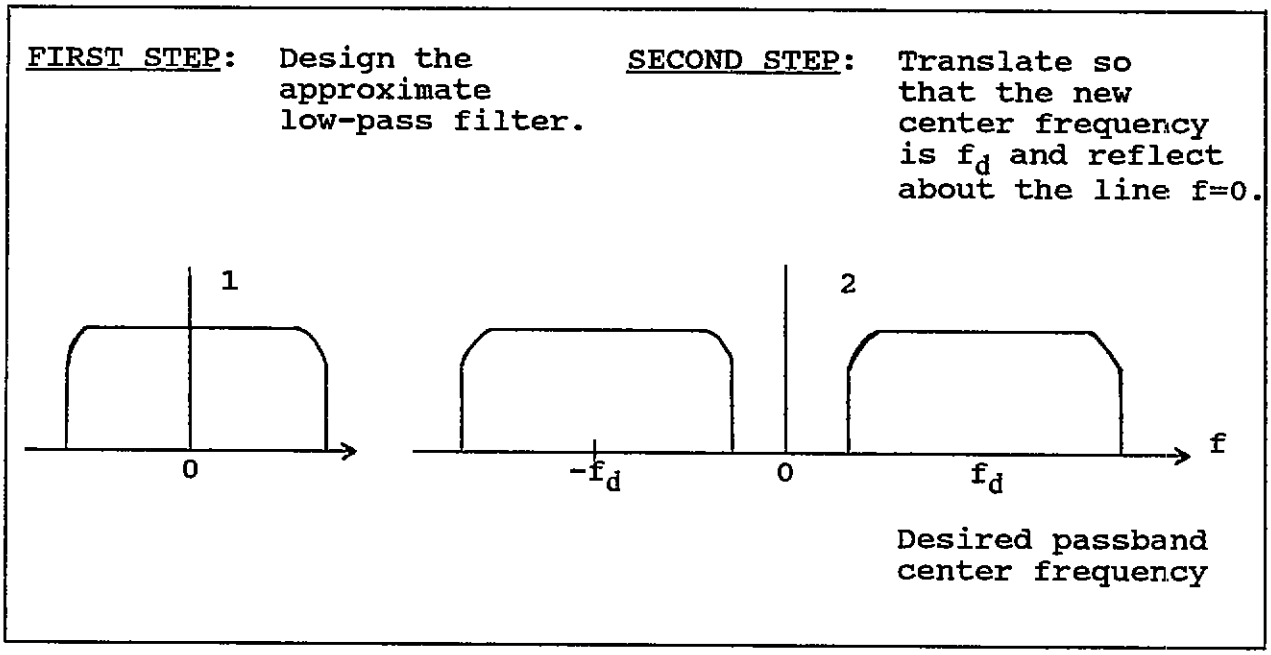

Figure 4-3. Method of converting a low-pass filter with shallow roll-off into a band-pass filter.

4.4 Band-Rejection Filters

To remove a particular band of frequencies from the data and, at the same time, to preserve the lower and higher frequencies, a band-rejection filter can be constructed. The band-rejection filter is the opposite of the band-pass filter. There

are various methods of constructing a band-rejection filter from a band-pass filter. Each method is analogous to an approach discussed in paragraph 4.2 in obtaining a high-pass filter from a low-pass filter.

The frequency response of the band-rejection filter can be obtained by subtracting the band-pass response from unity as shown in figure 4-4. In the time domain, the band-rejection results can be obtained by band-pass filtering and then subtracting the band pass-filter results from the raw data.

Another approach is to construct band-rejection weights and then filter the data directly in a one-step operation. The weights are constructed by

$$W_{R,k} = \delta_k - W_{B,k} \tag{4-8}$$

where

$W_{B,k}$ are the band pass weights (4-9)

and

$$\delta_k = \begin{cases} 1 \text{ if } k = 0 \\ 0 \text{ if } k \neq 0 \end{cases} \tag{4-10}$$

Filtering can be accomplished by convolving the weights of the band-rejection filter with the input data to obtain the filter.

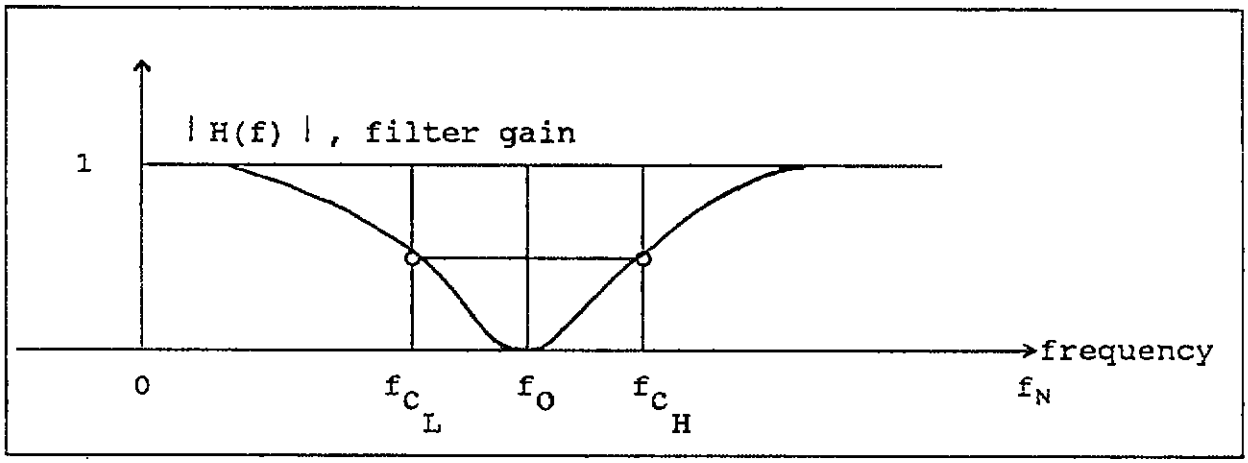

Figure 4-4. Frequency response of band-rejection filter.

4.5 Phase-Shift Filters

All categories of digital filters have a phase-shift curve which characterizes the amount of phase shift the filter induces for each frequency. It is sometimes desirable to construct a filter which will shift the phase of a signal by a constant amount over a narrow band of frequencies while maintaining constant amplitude. For example, a sinusoidal signal with amplitude $\propto$ and phase ϕ can be resolved into two components with amplitude $\propto \cos \phi$ and phase 0° and amplitude $\propto \sin \phi$ and phase 90°. In this case, a filter is constructed which produces a phase shift of 90° and combines the output of the filter with the unshifted data in the appropriate manner to produce any desired resultant phase shift. A phase lead of 90° and no amplitude change corresponds to frequency response equal to $\sqrt{-1}$. An ideal digital-phase shifter maintains this value for all frequencies between zero and the Nyquist frequency.

4.6 Filter Combinations

Any or all of the filters discussed in this chapter can be combined to form filters which perform more complex tasks. In any case, whatever effects are wanted can be accomplished in the time domain by applying the filters independently in series or by getting a one-step complex filter by convolving the filter weights together in the time domain. On the other hand in the frequency domain, the frequency response curves can be multiplied, added, or subtracted to give whatever response is desired, then transformed to the time domain via the appropriate use of the inverse Fourier transformation.

One special combination filter is referred to as a "comb filter." This filter consists of a series of n narrow band-pass filters having a frequency response curve with n narrow frequency bands at equal spaces between zero and the Nyquist frequency, which was discussed in subparagraph 3.7.3. This type of filter is constructed by convolving the raw data with the weights of n narrow band pass filters that can be accomplished in a one-step operation. (A similar one-step operation is discussed in paragraph 4.4, in converting a band-pass filter into a band-rejection filter.) The filter is then usually used to aid in determining significant frequencies and cut-off levels for other low-pass or high-pass filters. The Fast Fourier Transform is usually considered a better tool for this analysis than the comb filter.

CHAPTER 5

FREQUENCY DOMAIN FILTER DESIGN

There have been many filters developed for digital signal processing in the frequency domain and various ways to express these filters. In this chapter, a synopsis of the filters is given along with their various modes of expression. For recursive filters, methods of conversion from the analog form to the digital form are given as well. Also discussed are the advantages and disadvantages of using each type of filter, mode of expression, and method of conversion. Since the approach to designing infinite impulse response (IIR) (recursive) and finite impulse response (FIR) (nonrecursive) filters, described earlier in this document, are completely different, the chapter is divided into two parts: IIR and FIR filter design.

5.1 Infinite Impulse Response (IIR) Filter Design

The IIR filters have been extensively analyzed in the analog form. Consequently, digital IIR filters are usually described first in analog form and then converted to digital form.

The problem of approximating an IIR filter involves a complex transfer function of the frequency ω. The transfer function can be written as

$$H(z) = |H(z)| e^{iB(z)}, \tag{5-1}$$

where $z=e^{i\omega}$, and $B(z)=B(e^{i\omega})$ is the phase angle, which can be defined as

$$B(e^{i\omega}) = \arctan \frac{Im(H(e^{i\omega}))}{Re(H(e^{i\omega}))}. \tag{5-2}$$

Thus both the magnitude and the phase angle (also called the phase response) are examined to determine the nature of the filter approximations.

Advantages to using an IIR filter are that it (1) can be expressed in closed form and thus can generally be computed more efficiently, (2) does not require powerful computational facilities to be calculated, and (3) achieves a superior amplitude response. An undesirable consequence of using an IIR filter is that it yields a nonlinear phase response. More on the linear phase characteristic is covered in paragraph 5.2.

5.1.1 Modes of IIR Filter Expression

Recall from chapter 3 that an IIR filter can be expressed as

$$y(k) = \sum_{i=0}^{N} a(i)\, x(k-i) + \sum_{i=1}^{M} b(i)\, y(k-i) \qquad (5\text{-}3)$$

Three popular modes of expression for the transfer function of an IIR filter are

(1) the direct form,

(2) the cascade form, and

(3) the parallel form.

The forms assume those names because when they are illustrated in network diagrams, they are depicted in the forms mentioned. In the following subparagraphs, the mathematical forms and illustrations are given as well as the advantages and disadvantages.

5.1.1.1 Direct Form

As mentioned in paragraph 3.6, the z-transform of a weight function, that is, the transfer function of an IIR filter can be written as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=0}^{N} a_k z^{-k}}. \qquad (5\text{-}4)$$

The network design illustration for this equation is known as the direct form realization of the IIR filter, which is expressed by equation (5-3). This illustration is given in figure 5-1 and is obtained from reference 56.

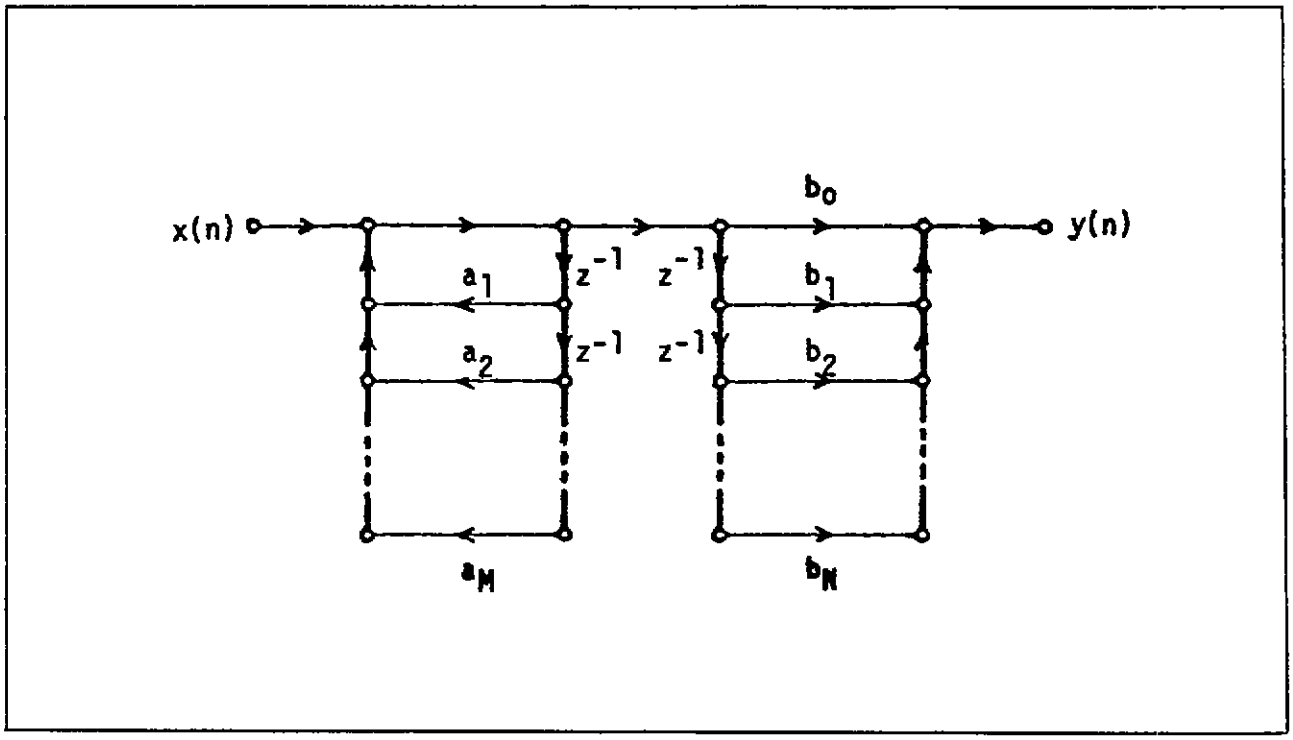

Figure 5-1. Direct form realization of IIR filter.

The advantages of this form are that the direct relation between the IIR filter equation (5-3) and the transfer function equation (5-4) is easily seen, and it is easy to formulate equation (5-4) given equation (5-3). The disadvantage of this form is that coefficients a_k and b_k are very sensitive to discretization errors. These errors will affect the accuracy and possibly the stability of the calculations. This form is generally not recommended for filters of order three or higher.

5.1.1.2 Cascade Form

Equation (5-4) can be rewritten as

$$H(z) = A \frac{\prod_{k=1}^{M_1} (1 - g_k z^{-1}) \prod_{k=1}^{M_2} (1 - h_k z^{-1})(1 - h_k^* z^{-1})}{\prod_{k=1}^{N_1} (1 - c_k z^{-1}) \prod_{k=1}^{N_2} (1 - d_k z^{-1})(1 - d_k^* z^{-1})} \qquad (5\text{-}5)$$

where $M_1 + M_2 = M$ and $N_1 + N_2 = N$. The numbers g_k and c_k are the real zeroes and poles of the transfer function. The numbers h_k, h_k^*, d_k, and d_k^* are the complex conjugate zeroes and poles of the transfer function. (A is a real number.)

This equation can be written as

$$H(z) = A \sum_{k=1}^{[(N+1)/2]} \frac{1 + {}_{1k}z^{-1} + {}_{2k}z^{-2}}{1 + {}_{\alpha 1k}z^{-1} + {}_{\alpha 2k}z^{-2}} \qquad (5\text{-}6)$$

(where $[(N+1)/2]$ is the greatest integer less than or equal to $(N+1)/2$). Assume that $M \leq N$. If $N \leq M$, replace N in equation (5-6) with M.

The network design illustration for this equation is known as the "cascade form realization" of the IIR filter, which is expressed by equation (5-3). This illustration is given in figure 5-2 and is obtained from reference 56.

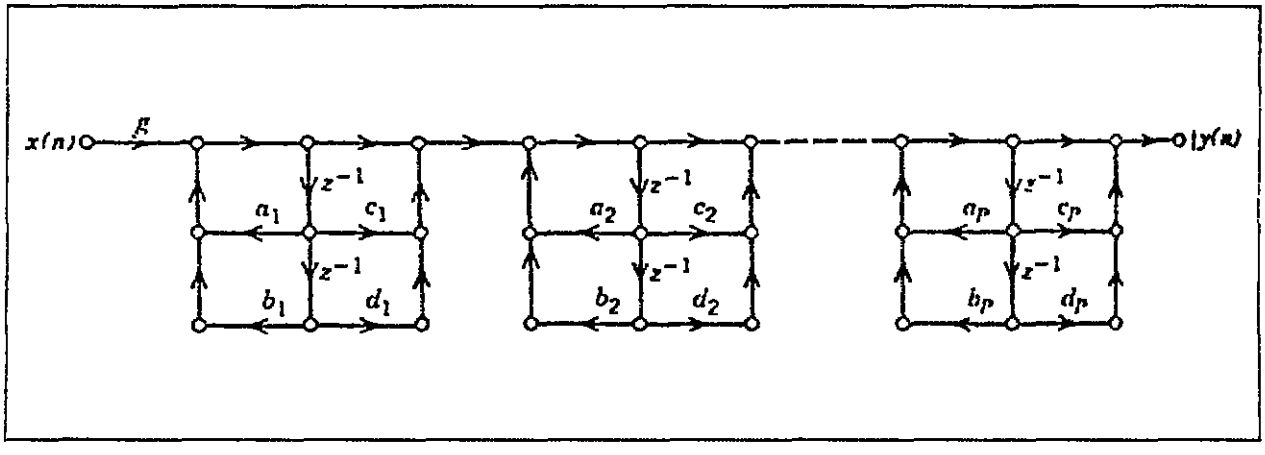

Figure 5-2. Cascade form realization of the IIR filter.

The advantages of this form are

(1) the zeroes (roots) of H(z) are easy to find,

(2) it is convenient to use for filters whose parameters are to be computed and changed in real time, and

(3) the poles (roots of the denominator) are easy to find.

The poles are important because they determine the stability of a filter. The filter is stable if and only if the poles are inside the unit circle. If a pole is on the unit circle, other criteria have to be used to determine stability.

A disadvantage of this form is that it is heavily subject to underflow and overflow problems.

5.1.1.3 Parallel Form

Equation (5-4) can be rewritten in terms of a partial fraction expansion as

$$H(z) = \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})} + \sum_{k=1}^{M-N} C_k z^{-1} \qquad (5\text{-}7)$$

The first two terms in equation (5-7) can be combined so that the equation becomes

$$H(z) = \sum_{k=1}^{M-N} C_k z^{-1} + \sum_{k=1}^{[(N+1)/2]} \frac{T_{0k} + T_{1k} z^{-1}}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}} \qquad (5\text{-}8)$$

The network design illustration for this equation is known as the "parallel form realization" of the IIR filter, which is expressed by equation (5-3). This illustration is given in figure 5-3 and is obtained from reference 55.

An advantage of this form is that the problems mentioned for the direct and cascade forms are normally not encountered. A disadvantage is that while pole locations can be easily located (as in the cascade form), the zeroes cannot.

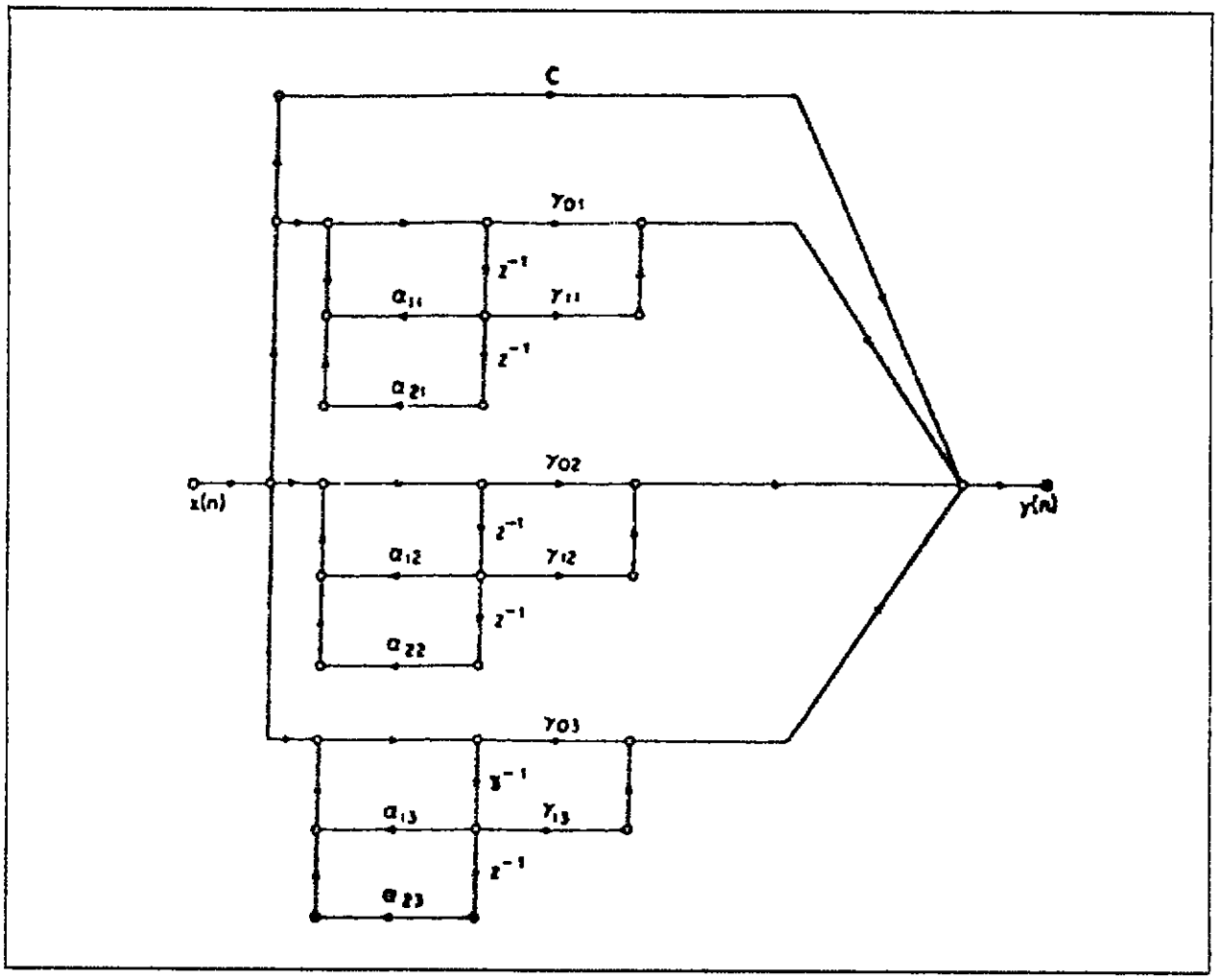

Figure 5-3. Parallel-form realization with the real and complex poles grouped in pairs.

5.1.2 Converting From Analog to Digital

The filters to be discussed in subparagraph 5.1.3 were originally derived in the analog form. Since these filters were discovered, a need developed to use frequency domain filters in the digital form. Because the art of analog filter design is highly advanced, it is considered advantageous to adopt the design procedures developed for the analog form to convert to the digital form. In this section four methods of analog-to-digital conversion are given along with their advantages and disadvantages.

The two desirable properties of any conversion method are

(1) The frequency axis in the "s plane" (the analog plane) be mapped to the unit circle in the "z plane" (the digital plane), ensuring that there is a one-to-one mapping between frequencies in the s plane and frequencies in the z plane. If this property is satisfied, the "frequency selective properties" of the analog filter are said to be preserved. With a conversion method that has this characteristic, the digital filter should filter through all and only those frequencies that the analog filter filters through.

(2) The left half of the s plane {Re[s]<0} be inside the unit circle in the z plane ($|z|<1$), which ensures the preservation of filter stability.

All the recommended conversion methods described in these subparagraphs have at least the second property.

5.1.2.1 Impulse Invariance

The main idea of the impulse invariance method is to preserve the impulse response when converting from analog to digital. That is, the purpose is to ensure

$$h(n) = h_a(nT) \tag{5-9}$$

where T is the sampling period, h is the digital impulse response, and h_a is the analog impulse response. In other words, the characteristic property of this transformation is that the impulse response of the resulting digital filter is a sampled version of the impulse response of the analog filter. Use the following method:

(1) set up analog transfer function H(s) in direct, cascade, or parallel form. (Generally, the parallel form is the most preferable, because it is easy to perform step (2) when using a table for transform pairs);

(2) obtain the inverse Laplace (or Fourier) transform of H(s), giving the analog impulse response function h(t);

(3) get your digital impulse response function h(n) using equation (5-9); and

(4) take the z-transform of h(n) to obtain your digital transfer function H(z).

Steps (2) and (4) are jointly justified, because it can be shown that the Laplace transform of the analog function h_a is related to the z-transform of the digital function h(t) by

$$H(z)\Big|_{z=e^{sT}} = (1/T)\sum_{k=-\infty}^{\infty} H_a(s + i(2\pi/T)k). \qquad (5\text{-}10)$$

With the z-transform, the relationship $z=e^{sT}$ is used in going from the s plane (for the analog function) to the z plane (for the digital function). Each horizontal strip of the left half of the s plane of width $2\pi/T$ is mapped into the unit circle in the z plane (see figure 5-4, obtained from reference 55). This multiple mapping may lead to aliasing (explained in chapter 3) because distinct frequencies from different strips in the s plane can be translated into one frequency in the z plane.

The occurrence of aliasing can be seen by noting that if $s = \sigma + i\pi$, then

$$z = e^{sT} = e^{\sigma T}(\cos\pi T + i*\sin\pi T)$$

$$= e^{\sigma T}(\cos(\omega + (2\pi n/T))T + i*\sin(\omega + (2\pi n/T))T) \qquad (5\text{-}11)$$

for any integer n.

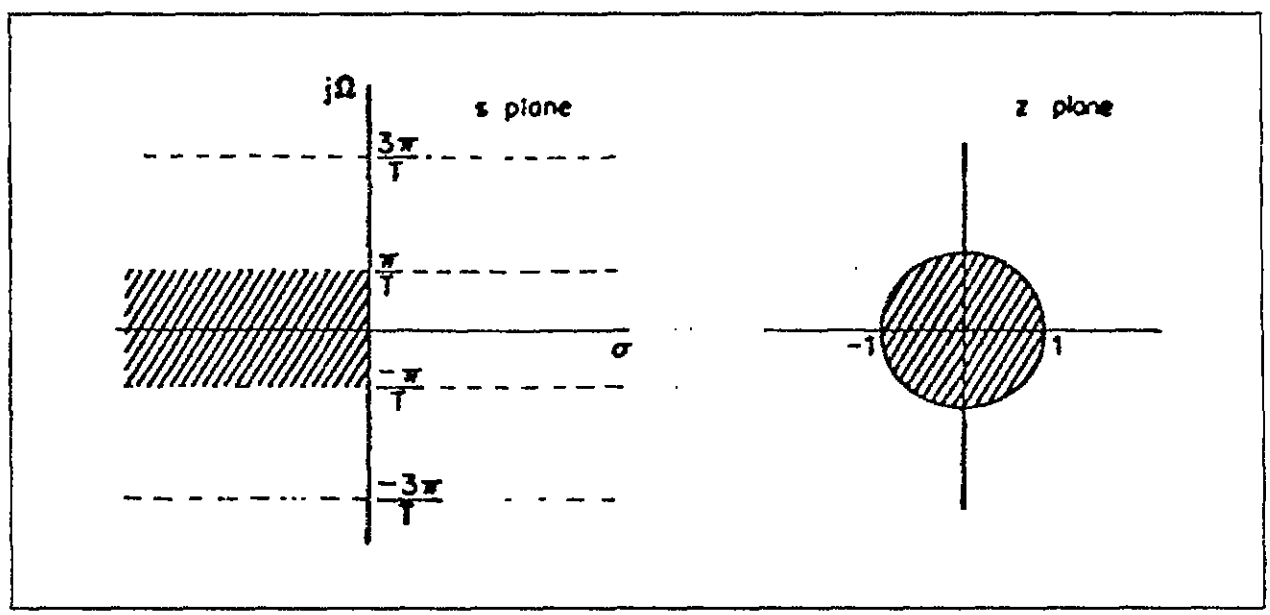

Figure 5-4. The mapping of a horizontal strip of width $2\pi/T$ in the s plane to the unit circle of the z plane, through the impulse invariance method.

A disadvantage in using this method is that it is subject to aliasing. This method works best when used with bandlimited filters such as low pass and band pass.

5.1.2.2 Bilinear Transformation

With the bilinear transformation, to go from the s plane (which represents the analog filter) to the z plane (which represents the digital filter), the following equation is used:

$$z = \frac{1 + (T/2)s}{1 - (T/2)s} \tag{5-12}$$

Likewise, s can be expressed in terms of z by

$$s = \frac{2}{T}\,\frac{1 - z^{-1}}{1 + z^{-1}} \tag{5-13}$$

That is, the transfer function of the digital filter, H(z), is set equal to the transfer function of the analog filter, H(s), where s is expressed in terms of z as in equation (5-13), or where z is expressed in terms of s as in equation (5-12). These equations were obtained by the procedure outlined next.

A digital filter is essentially a difference equation. For an analog filter, a differential equation is used. The analog transfer function is obtained by using Fourier or Laplace transforms. To obtain the bilinear transformation, integrate both sides of the differential equation and use a numerical approximation to the integral so that the integrated equation is expressed in discrete terms. Then take the z-transform of this equation to get the digital transfer function. When comparing the analog and digital transfer functions, it can be seen that s and z are related as stated in equations (5-12) and (5-13). Reference 55 gives the derivation of these equations in more detail.

Illustrated in figure 5-5, which was obtained from reference 55, is the relationship between s and z.

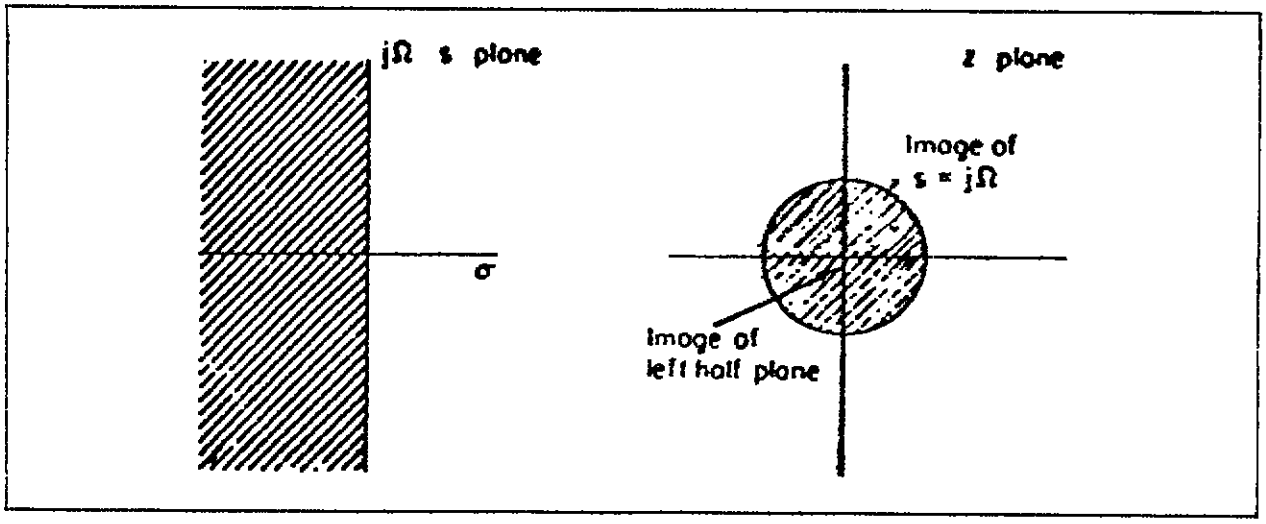

Figure 5-5. The mapping of the left half of the s plane to the unit circle of the z plane, using the bilinear transformation.

Reference 55 shows that the analog frequency Ω can be expressed in terms of the digital frequency ω as follows:

$$\Omega = (2/T)\tan(\omega/2) \qquad (5\text{-}14)$$

The graphic relationship between Ω and ω is given in figure 5-6, which was obtained from reference 54.

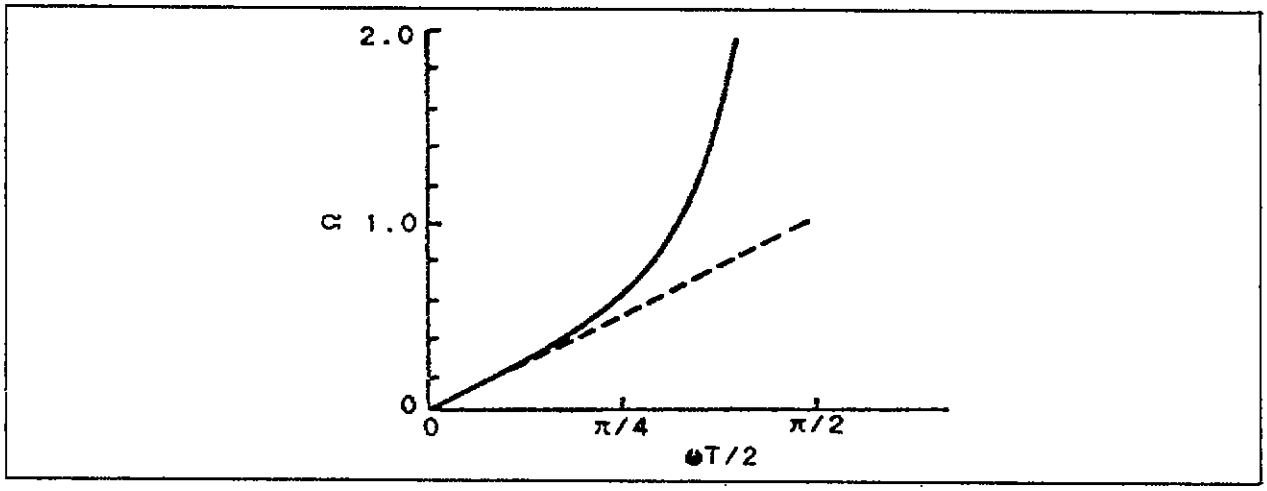

Figure 5-6. The relation between analog and digital frequency scales for the bilinear transformation.

As seen from the last two figures, the bilinear transformation maps an infinite range of frequencies onto the unit circle, representing a finite range of frequencies. This mapping, consequently, distorts the frequency scale.

The frequencies of interest in the s plane must be predistorted to make sure they will come out in the right places in the z plane. The predistortion formula is

$$\Omega = \frac{2}{T*\tan(\omega T/2)}. \qquad (5\text{-}15)$$

To sum up, this method will probably produce warping in the frequency axes. Frequencies in the s plane are mapped into frequencies in the z plane in a distorted manner. To make up for this distortion, the frequencies can be prewarped. As given in equation (5-15), this method satisfies property (1) of subparagraph 5.1.2. This method works best when used with nonband-limited filters such as band stop or high pass.

5.1.2.3 Direct Mapping of Differentials

There are three ways to employ the direct mapping of differentials: backward difference, forward difference, and generalized difference. Consider the following differential equation representing the analog filter:

$$\sum_{i=0}^{N} a_i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{M} b_i \frac{d^i x(t)}{dt^i} \qquad (5\text{-}16)$$

Equation (5-16) is then discretized by

$$\sum_{i=0}^{N} a_i \Delta_i[y(n)] = \sum_{i=0}^{M} b_i \Delta_i[x(n)] \qquad (5\text{-}17)$$

where $\Delta_i[y(n)]$ is the i^{th} difference defined by the recursion

$$\Delta_{i+1}[y(n)] = \Delta_1\{\Delta_i[y(n)]\} \qquad (5\text{-}18)$$

and by the initialization

$$\Delta_i[y(n)] = \begin{cases} (1/T)[y(n) - y(n-1)] & \text{backward difference} \\ (1/T)[y(n+1) - y(n)] & \text{forward difference} \\ \\ \frac{1}{T}\sum_{i=1}^{L} \alpha_i[y(n+i) - y(n-i)] & \text{generalized difference} \end{cases} \tag{5-19}$$

and $\Delta_i[x(n)]$ is defined in the same way. The factor α_i for the generalized difference is a constant of the user's choice.

Backward Differences: When using backward differences, make the replacement

$$\frac{dy}{dt} \longleftrightarrow \frac{y(n) - y(n-1)}{T} \tag{5-20}$$

which, in terms of the relationship between s and z, corresponds to

$$s = \frac{1 - z^{-1}}{T} \tag{5-21}$$

and

$$z = \frac{1}{1 - sT} \tag{5-22}$$

The relationship between s and z is illustrated in figure 5-7, (obtained from reference 54). As can be seen property (1) mentioned in subparagraph 5.1.2 (one-to-one frequency correspondence) is not satisfied and that property (2) (stability preservation) is satisfied.

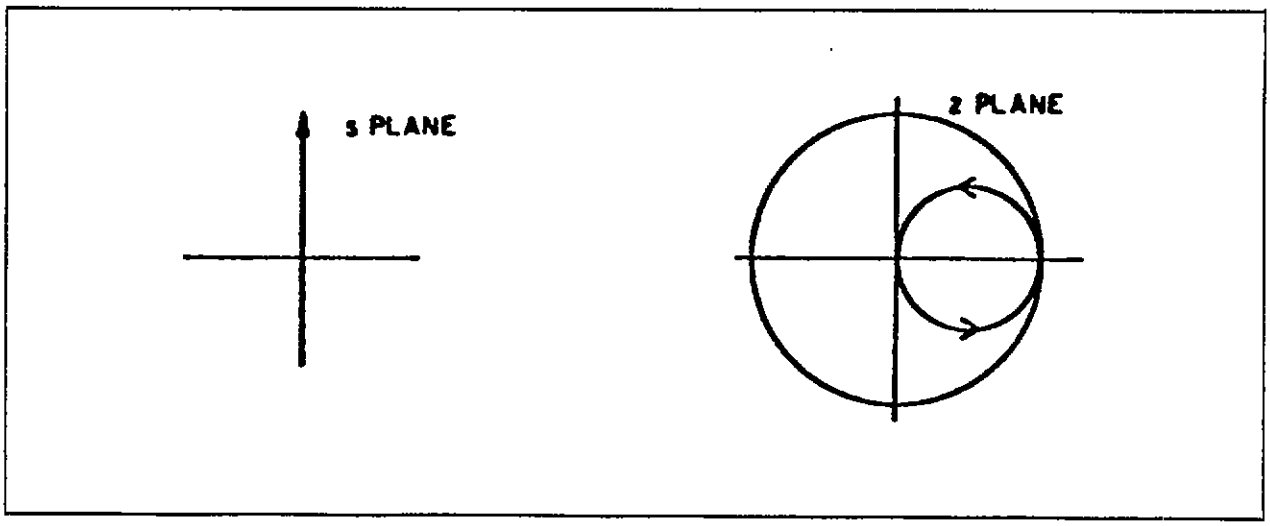

Figure 5-7. s plane to z plane mapping of jΩ axis for method of backward differences.

Reference 55 describes how property (1) becomes closer to being satisfied the higher the sampling rate. The high sampling rate required to make this technique adequate for analog-to-digital conversion is said to result in a very inefficient representation of the filter and the input signal. This technique is considered usable for low-pass filters only. An example for which this technique can be used is air flight control, where frequencies are normally at 100 Hz or less. An advantage of this method is the simplicity of the design.

Forward Differences: When using backward differences, we make the replacement

$$\frac{dy}{dt} \longleftrightarrow \frac{y(n+1) - y(n)}{T} \tag{5-23}$$

which, in terms of the relationship between s and z, corresponds to

$$s = \frac{z - 1}{T} \tag{5-24}$$

and

$$z = 1 + sT \tag{5-25}$$

The mapping of the frequency axis from the s plane to the z plane is illustrated in figure 5-8, as obtained from reference 54. As can be seen property (1) mentioned in subparagraph 5.1.2 (one-to-one frequency correspondence) is not satisfied. Also shown from either equation (5-16) or equation (5-17) property (2) (stability preservation) is not satisfied either. This method is not recommended.

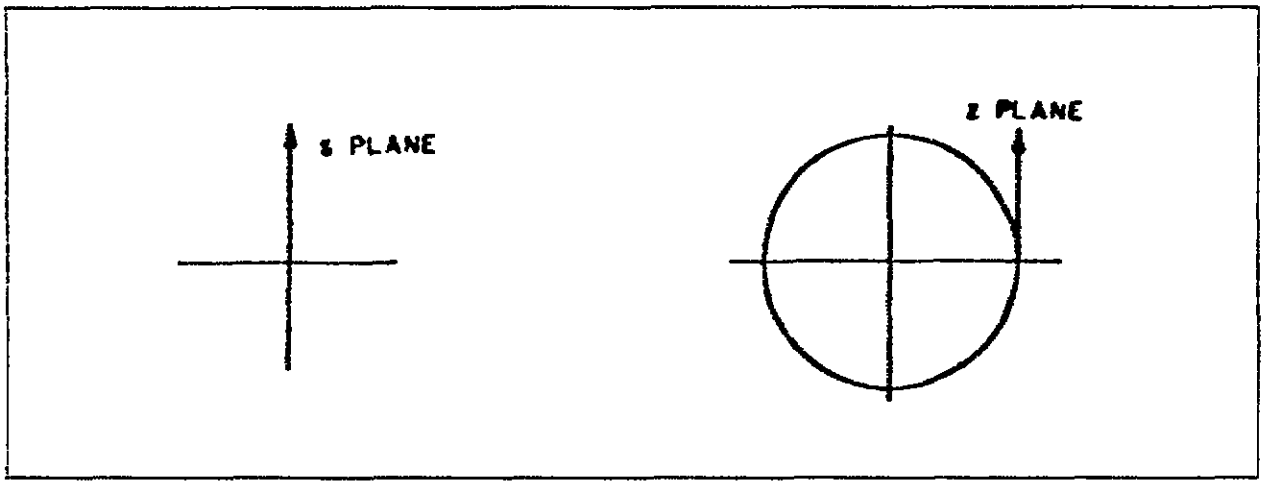

Figure 5-8. s plane to z plane mapping of jΩ axis for method of forward differences.

Generalized Differences: As can be seen from equation (5-19), this method uses higher-order differences to replace lower-order differentials. The mapping between the s plane and the z plane for this method is

$$s = \frac{1}{T} \sum_{i=1}^{L} \alpha i (z^{i} - z^{-i}) \qquad (5\text{-}26)$$

where L is the order of difference to be used. Reference 54 shows that with the proper choice of coefficients α_i, the frequency axis in the s plane is mapped monotonically to the unit circle in the z plane, thereby satisfying property (1) in subparagraph 5.1.2. The mapping of equation (5-26) can be shown to be conformal. A conformal mapping preserves angles and relative locations of points from the domain to the range. (From this conformality, the left half of the s plane is mapped to the inside of the unit circle of the z plane, thereby, showing that property (2) is satisfied.)

This method is considered efficient and accurate, if the appropriate coefficients α_i can be found. It is normally difficult to determine these coefficients. As a result, other techniques for digitizing filters are sought.

5.1.2.4 Matched Z-Transformation

The matched z-transformation is motivated by the following facts, stated without proof:

(1) to maintain stability, all the poles of H(z), the transfer function in the z plane, must lie within the unit circle, and

(2) all poles and zeroes must either be real or occur in complex conjugate pairs.

As such, this method matches poles and zeroes in the s plane to poles and zeroes in the z plane. For a real pole or zero, say -a, the transformation is

$$s + a \longrightarrow 1 - z^{-1}e^{-aT} \qquad (5\text{-}27)$$

where T is the sampling period.

For complex conjugate poles or zeroes, say $a \pm bi$, the transformation is

$$(s+a-bi)(s+a+bi) =$$

$$(s+a)^2 + b^2 \longrightarrow 1 - 2z^{-1}e^{-aT}\cos(bT) + z^{-2}e^{-2aT} \qquad (5\text{-}28)$$

(Since b=0 in the real case, it is easily seen that equation (5-28) can be simplified to equation (5-27).) The continuous transfer function H(s) must be in factored form to apply the transformation. The advantages of this method are that it ensures stability (it was designed to) and that it is very easy to implement. One disadvantage is that it may lead to aliasing. For instance,

$(s+a)^2 + b^2$ yields the same transformation as $(s+a)^2 + (b+2\pi)^2$. (5-29)

This transformation is considered unsuitable where H(s) is an all-pole system; that is, H(s) has only poles and no zeroes. Quite often for this case, H(z) is an all-pole system that does not adequately represent the desired continuous system. Generally, the bilinear transformation and impulse invariance methods are said to be preferred over the matched z-transformation.

5.1.3 Low-Pass IIR Filters

Discussed next are three classes of low-pass IIR filters: the Butterworth, the Chebyshev, and the elliptic (or "Cauer") filters. Each of these filters takes the form

$$H(\omega)^2 = \frac{1}{1 + \epsilon^2 f_N(\omega)} \tag{5-30}$$

where H is the frequency response, ω is the frequency variable, N is the order of the filter, ϵ is a factor in the interval [0,1] that determines the height of the passband ripples, and f is an nth-order polynomial containing only odd or only even powers of ω. Bandpass, high-pass, and band-rejection filters can then be designed from any of the aforementioned low-pass filters using the methods described in chapter 4.

5.1.3.1 Butterworth Filters

Butterworth filters take the form

$$|H(\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2N}} \tag{5-31}$$

where ω_c is the cutoff frequency of the filter.

The Butterworth filter contains the following characteristics:

(1) It is defined by the property that the magnitude response $|H(\omega)|$ is maximally flat in the passband, meaning that the maximum number of derivatives, 2N - 1, of the squared magnitude function $|H(\omega)|^2$ are equal to zero at ω=0.

(2) $|H(\omega)| = 1/\sqrt{2}$ at the cutoff frequency ω_c since, when attempting to simulate the ideal low-pass filter, it is desired that $\lim_{N\to\infty} |H(\omega)| = 1$ for all ω in ω

$[-\omega_c, \omega_c]$. ($\lim_{N\to\infty} |H(\omega)| = 1$ for all $|\omega| < \omega_c$.)

(3) It is computationally and conceptually simpler than the Chebyshev and elliptic filters.

(4) The higher N is, the better this filter simulates the ideal low-pass filter.

(5) The magnitude of the frequency response is monotonic in both the passband and the stop band.

Illustrations of the Butterworth filter, obtained from reference 55, can be seen in figure 5-9.

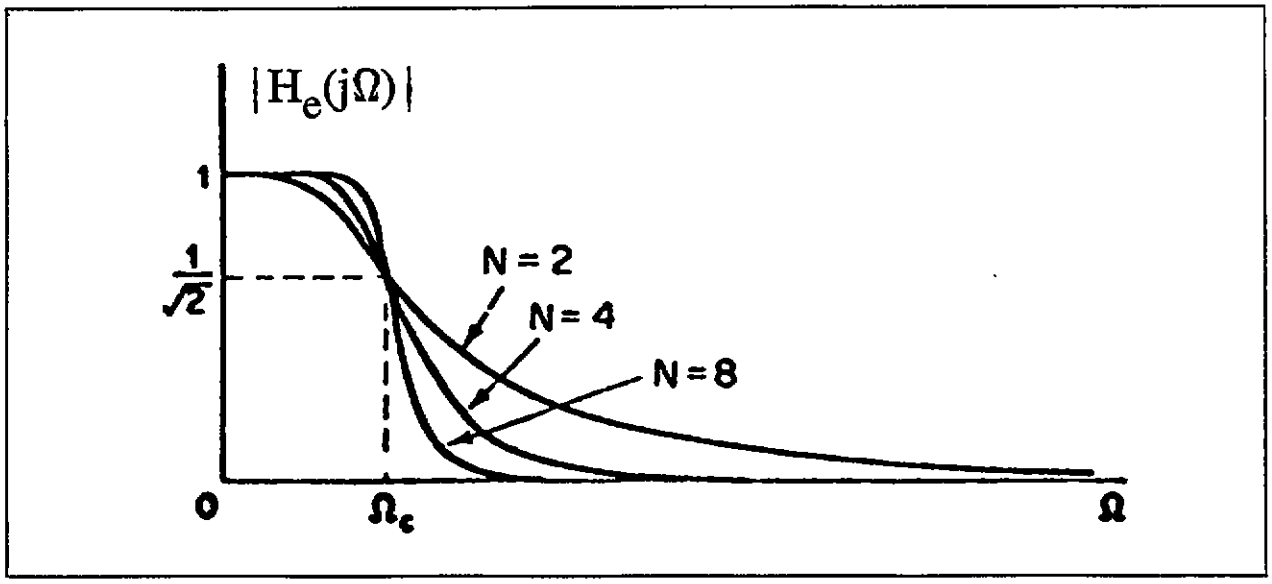

Figure 5-9. Dependence of Butterworth magnitude characteristic on the order N.

5.1.3.2 Chebyshev Filters

Chebyshev filters take the form

$$|H(\omega)|^2 = \frac{1}{1 + \epsilon^2 V_N^2(\omega/\omega_c)} \qquad (5\text{-}32)$$

where $V_N(x)$ is the nth order Chebyshev polynomial defined by

$$V_N(x) = \cos(N\cos^{-1}x). \qquad (5\text{-}33)$$

(For example, for N=2, $V_N(x)=V_2(x)=2x^2-1$.) (5-34)

The Chebyshev filter contains the following characteristics:

(1) It distributes uniformly the inaccuracy in simulating the ideal LP filter in either the passband or the stop band (but not both). That is, it produces an equiripple curve either in the passband (about the line $|H(\omega)|^2=1$) or in the stop band (about the line $|H(\omega)|^2=0$).

(2) It usually leads to a lower-order polynomial than does the Butterworth filter to accomplish the same result.

There are two types of Chebyshev filters. Chebyshev Type I filters are equiripple in the passband and monotonic in the stop band. Chebyshev Type II filters (sometimes called inverse Chebyshev filters) are monotonic in the passband and equiripple in the stop band. Figure 5-10, obtained from reference 54, illustrates Type I and Type II Chebyshev filters of odd and even orders. (The A near the vertical axes is what reference 54 uses at the parameter related to stop-band loss.)

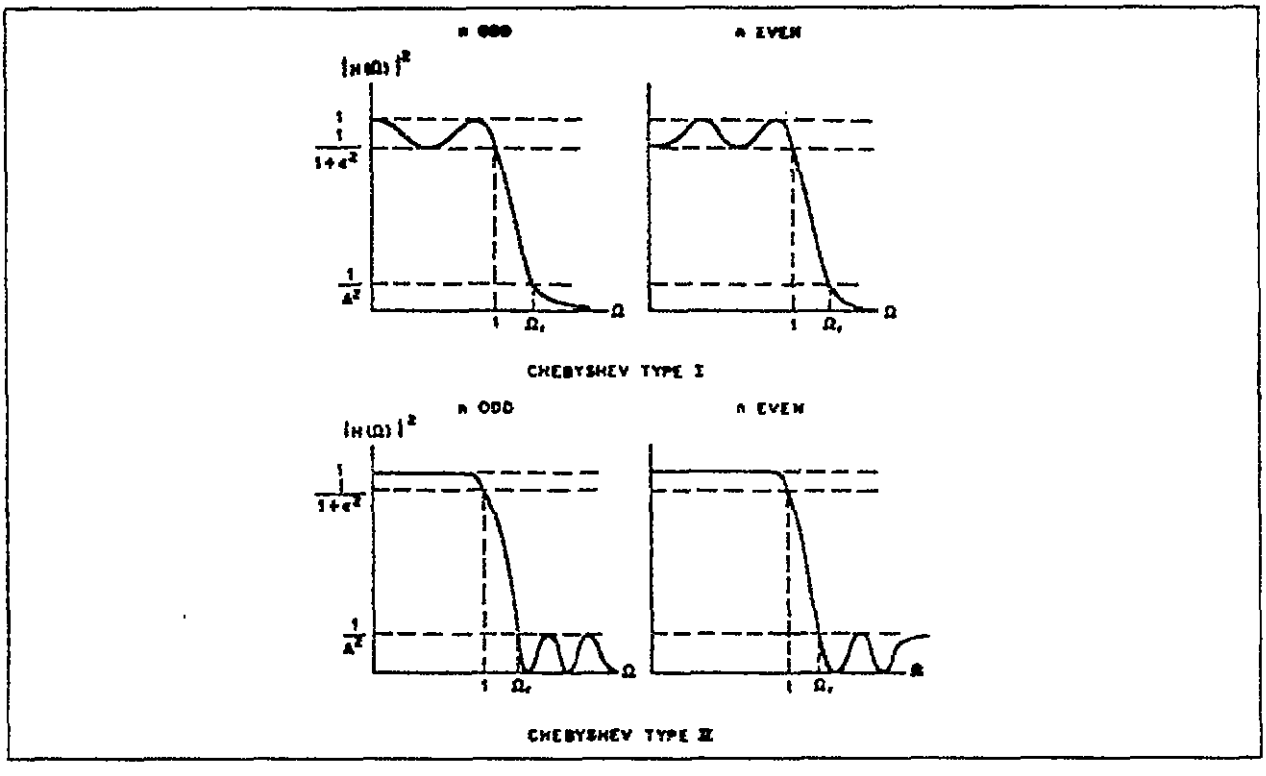

Figure 5-10. Type I and II Chebyshev filters of odd and even orders.

5.1.3.3 Elliptic Filters

Elliptic (or Cauer) filters take the form

$$|H(\omega)|^2 = \frac{1}{1 + \epsilon^2 U_N^2(\omega, L)} \qquad (5\text{-}35)$$

where $U_N(\omega,L)$ is a Jacobian elliptic function. Discussion of this function is highly intricate and is beyond the realm of this paper. Those interested in studying this function are referred to reference 58. Those interested in further studying the design of elliptic filters are referred to references 59, 60, and 61.

The elliptic filter contains the following characteristics:

(1) It distributes uniformly the inaccuracy in simulating the ideal LP filter in both the passband and the stop band. In other words, it produces an equiripple curve in the passband (about the line $|H(\omega)|^2=1$) and in the stop band (about the line $|H(\omega)|^2=0$).

(2) It yields a smaller transition band than does the Chebyshev. (Note that the Butterworth filter does not yield a transition band.)

Figure 5-11, obtained from reference 55, illustrates the elliptic filter.

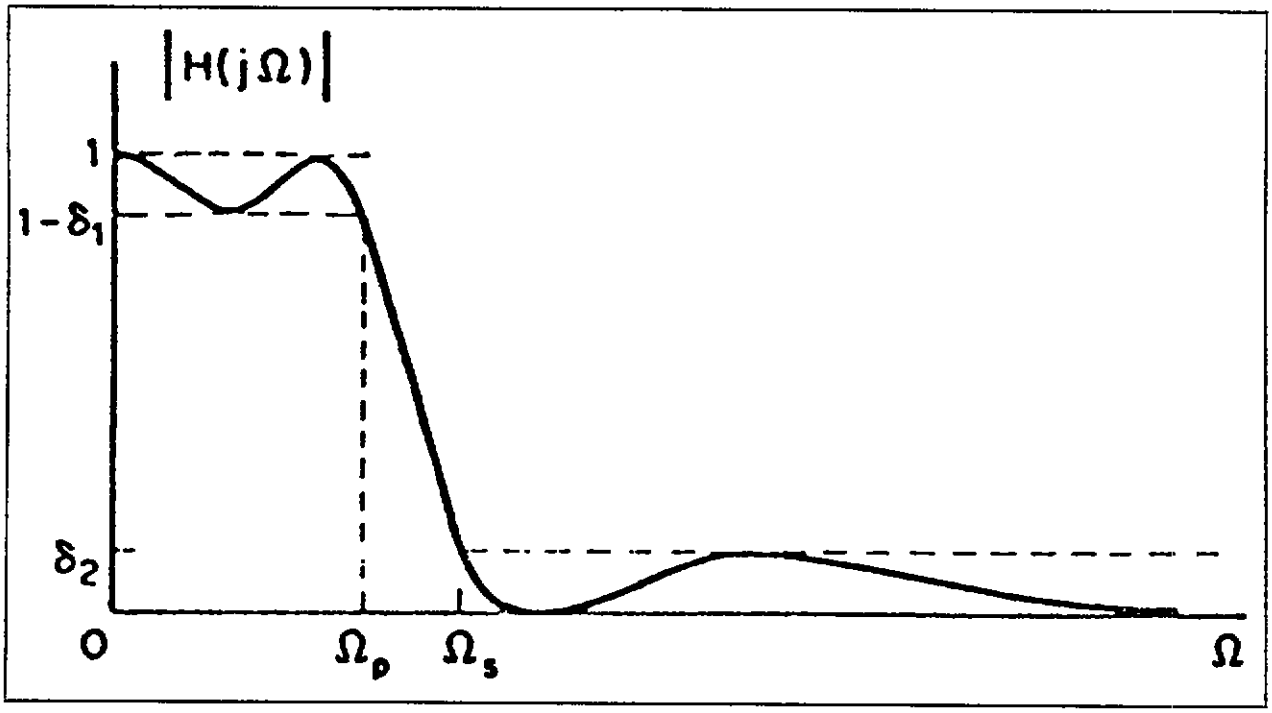

Figure 5-11. Equiripple approximation in both passband and stop band.

5.2 Finite Impulse Response (FIR) Filter Design

The FIR filters were briefly described in paragraphs 1.7 and 1.8. In this section, modes of expression for FIR filters and FIR filter design techniques are given. A disadvantage in implementing an FIR filter is that it cannot be expressed in closed form. Instead, it must be calculated through iterative procedures, and thus generally requires a great deal of computational time and the use of powerful computational facilities.

In many signal-processing applications, phase relations are important and must not be disturbed by filtering. For such purposes, a zero phase shift would be ideal; in practice, however, a filter whose phase shift is proportional to frequency is generally used. Such a filter is called linear phase. An advantage in using FIR filters is that they can have exact linear phase. Referring to figure 3-6, it has been found that the most favorable conditions for an FIR design are large values of δ_1, small values of δ_2, and large transition widths.

5.2.1 Modes of FIR Filter Expression

An FIR filter can be expressed as

$$y(k) = \sum_{i=0}^{N} a(i)\, x(k-i) \qquad (5\text{-}36)$$

There are four popular modes of expression for the transfer function of an FIR filter. They are

(1) the direct form,

(2) the cascade form,

(3) the frequency sampling form, and

(4) the linear-phase form.

The first two forms assume those names because when they are illustrated in network diagrams, they are depicted in the forms mentioned. The direct and cascade FIR forms may be derived from the corresponding IIR forms by simply omitting the pole-producing portions of the IIR forms. (The second term on the right-hand side of equation (5-3) generates the pole-producing terms in the IIR modes of expression.) The advantages and disadvantages of the direct and cascade forms are essentially the same for FIR filters as they are for IIR filters. As a result, it will be sufficient just to give their forms in the following subparagraphs.

The frequency sampling form for FIR filters is a type of parallel form of IIR filters but is derived in an entirely different way from the parallel form. The fact that FIR filters can have exact linear phase is used for the linear-phase form. When given its form, it will be easy to see how it can produce considerable savings in computations.

5.2.1.1 Direct Form

The direct form of the transfer function for FIR filters can be found to be

$$H(z) = \sum_{k=0}^{N} a_k z^{-k} \tag{5-37}$$

5.2.1.2 Cascade Form

The cascade form for the transfer function of FIR filters is

$$H(z) = \frac{Y(z)}{X(z)} = a_0 \prod_{i=1}^{K} H_i(z) \tag{5-38}$$

where $H_i(z)$ is either a second-order cascade section, that is,

$$H_i(z) = 1 + a_{1i} z^{-1} + a_{2i} z^{-2} \tag{5-39}$$

or a first-order cascade section, that is,

$$H_i(z) = 1 + a_{1i} z^{-1} \tag{5-40}$$

and K is the integer part of (N+1)/2, where N is the order of the filter.

It should mention here that if linear-phase filters are realized in this form, sensitivity to discretization errors in the coefficients will be less, but the errors may destroy the phase linearity.

5.2.1.3 Frequency Sampling Form

As mentioned in subparagraph 5.2.1, this form is a type of parallel form but is derived differently from the parallel IIR form. The parallel IIR form was derived from a partial-fraction expansion of the transfer function. Since partial-fraction expansions are based on poles, this approach cannot be considered for FIR filters. This form is derived by a design technique called the frequency sampling design technique and is explained in subparagraph 5.2.2.2.

5.2.1.4 Linear-Phase Form

Many applications require filters whose phase response is linear with frequency. It can be shown that linear-phase FIR filters have a symmetrical impulse response, that is, for an N-order filter

$$h(n) = h(N-1-n) \tag{5-41}$$

This symmetry requirement leads to certain economies in implementation. In an FIR filter, the h's correspond one-to-one with the a's of equation (5-37). Because of the symmetry in h(n), equation (5-37) becomes

$$y(n) = a_0x(n) + a_1x(n-1) + \ldots + a_1x(n-N-1) + a_0x(n-N) \tag{5-42}$$

which can be written for N even, as

$$y(n) = \sum_{i=0}^{N/2 - 1} b_n[x(n-1)+x(n-N+i)] \tag{5-43}$$

or for N odd, as

$$y(n) = b_{N/2}x(n - N/2) + \sum_{1=0}^{(N-1)/2 - 1} b_n[x(n-1)+x(n-N+i)] \tag{5-44}$$

Since the number of terms in the sum is reduced by approximately one half, considerable economies in computation are achieved. Because of the symmetry in this form, errors in quantizing coefficients, that is, discretization errors of the coefficients, will not disturb the linear-phase characteristic although performance but may still degrade performance.

5.2.2 FIR Filter Design Techniques

The techniques covered in this section are

(1) windowing,

(2) frequency sampling design, and

(3) equiripple design.

There are many types of windowing and some will be briefly described here. The second two techniques are computer-aided design techniques and require a great deal of iterative calculation.

5.2.2.1 Windowing

It is natural to design a filter by first simulating the ideal low-pass filter, defined in equation (3-16). In doing so, the impulse response h(n) should meet these two conditions: (a) it is finite, and (b) it is causal. In particular, that $h(n) = 0$ for $n < 0$.

The impulse response is the inverse Fourier transform of the transfer function. The impulse response of the ideal low-pass filter transfer function can be found to be

$$h(n) = \frac{\sin(\omega_c n)}{n} \qquad (5\text{-}45)$$

Neither condition (a) or (b) above is met. The ideal low-pass filter and h(n) are shown in figures 5-12a and 5-12b, obtained from reference 56.

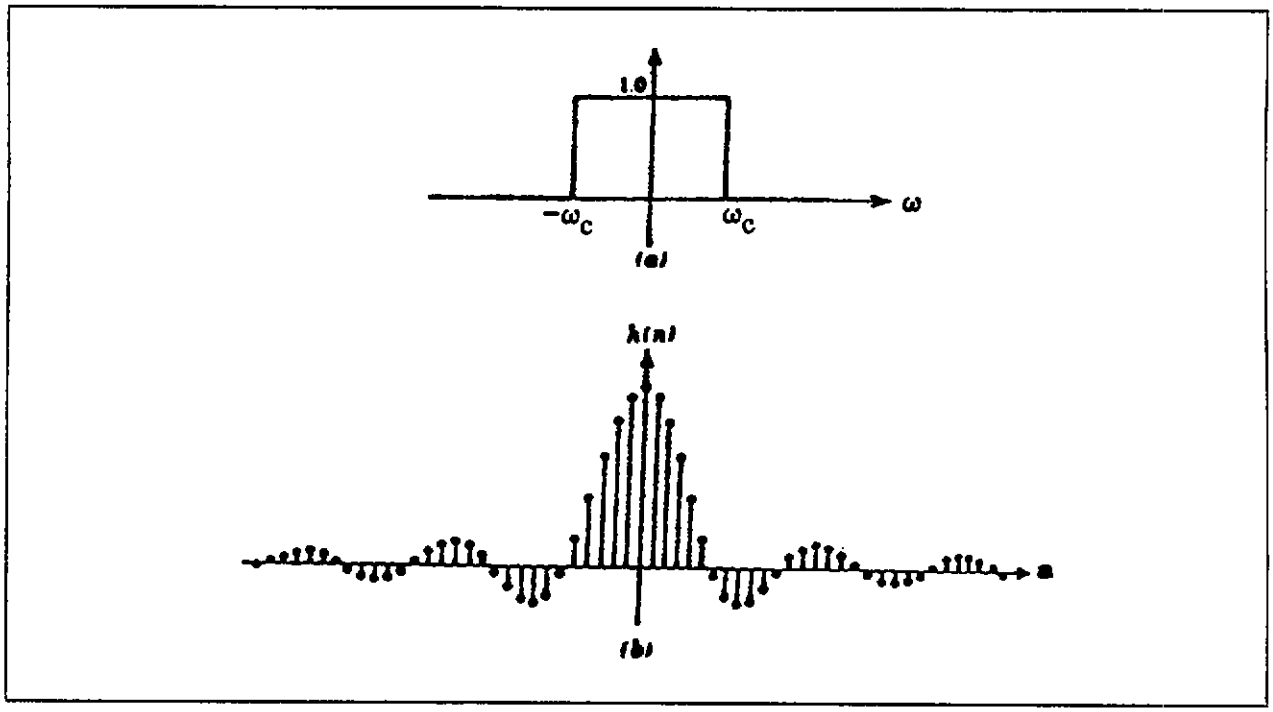

Figure 5-12. (a) Ideal low-pass filter characteristic with cutoff at ω_c and (b) impulse response corresponding to ideal low-pass filter.

A solution to meeting conditions (a) and (b) is to find a finite, causal approximation to h(n) which can be done by windowing the impulse response, (by truncating it for | n | greater than some cutoff time) and by shifting the response in time until the system is causal. The two steps are shown in figures 5-13a and 5-13b, obtained from reference 56.

As explained towards the end of paragraph 3.3, truncating the impulse response would result in a boxcar function with ripples (see figure 5-13c). Three deviations from the ideal low-pass filter emerge:

(1) the passband response is no longer flat but shows ripples that steadily increase in amplitude until the cutoff frequency,

(2) the stop band response is no longer zero, and

(3) the transition between passband and stop band is no longer abrupt.

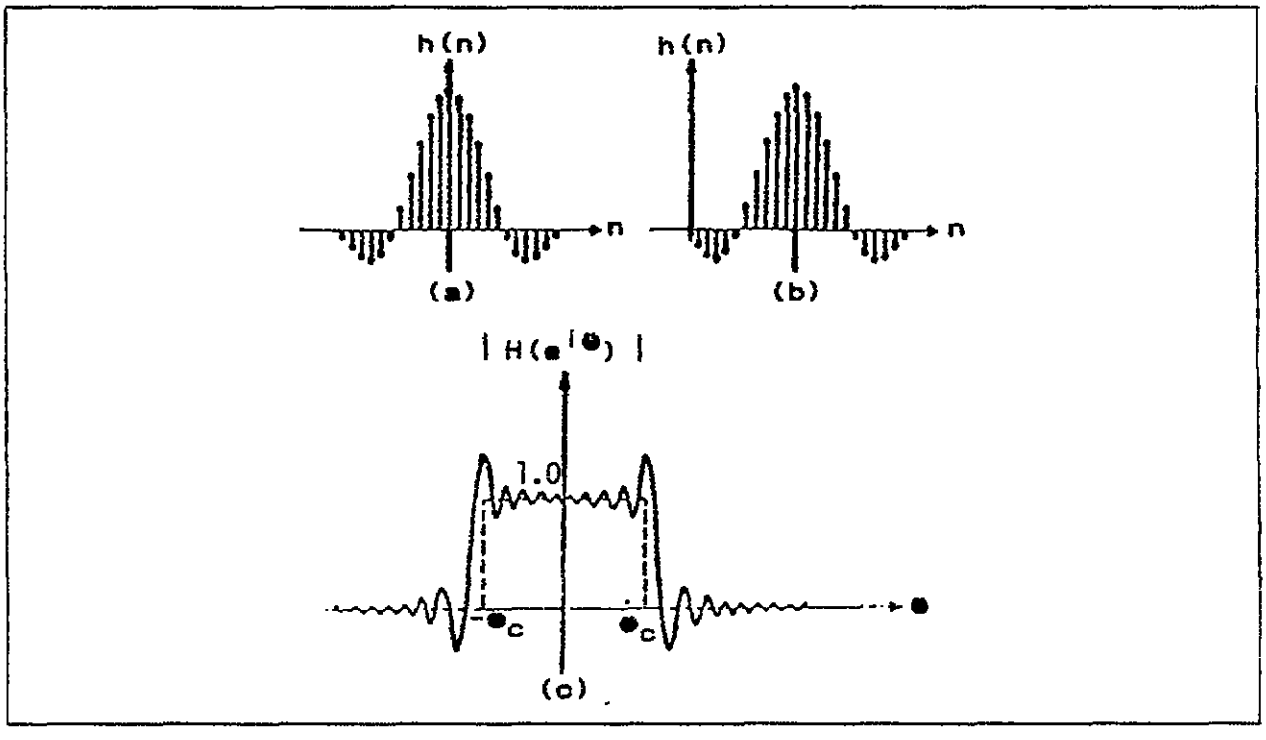

Figure 5-13. (a) Truncated version of impulse response in figure 5-12, (b) Truncated response shifted so as to make system causal, and (c) Filter frequency response resulting from truncation of impulse response.

To minimize these ripples, multiply the original infinite-impulse response by a windowing function (other than the rectangular function that was initially introduced). There is no finite window function whose transform has no side lobes, but functions can be found whose transforms have very small side lobes. If one of these functions is used, the ripples in the frequency response will be correspondingly reduced.

Some of the best-known windowing functions are listed in table 5-1 and shown in figure 5-14, along with their transforms. (The table and figure are obtained from reference 56.)

TABLE 5-1. COMMON WINDOWING FUNCTIONS

Name	Description[a]
Rectangular	$w(k) = 1$
Fejer-Bartlett	$w(k) = 1 - \lvert 2k/N \rvert$
Hanning	$w(k) = (1 + \cos \pi k/N)/2$
Hamming	$w(k) = 0.54 + 0.46 \cos \pi k/N$
Kaiser	$w(k) = \dfrac{I_0\left[N\theta_a\sqrt{1-(k/N)^2}\right]}{I_0[N\theta_a]}$

[a] For all windows, $w(k) = 0$ for $\lvert k \rvert > N$.

For the Kaiser window, I_0 is the zeroth-order Bessel function and θa is a constant that specifies a frequency response tradeoff between peak height of the side lobe ripples and the width or "energy" of the main lobe. (The Fejer-Bartlett window is also called the triangular window.) Reference 55 also mentions the Blackman window, which is $\omega(n) = 0.42 - 0.5\cos(2\pi n/(N-1)) + 0.08\cos(4\pi n/(N-1))$, $0 \le n \le N-1$.

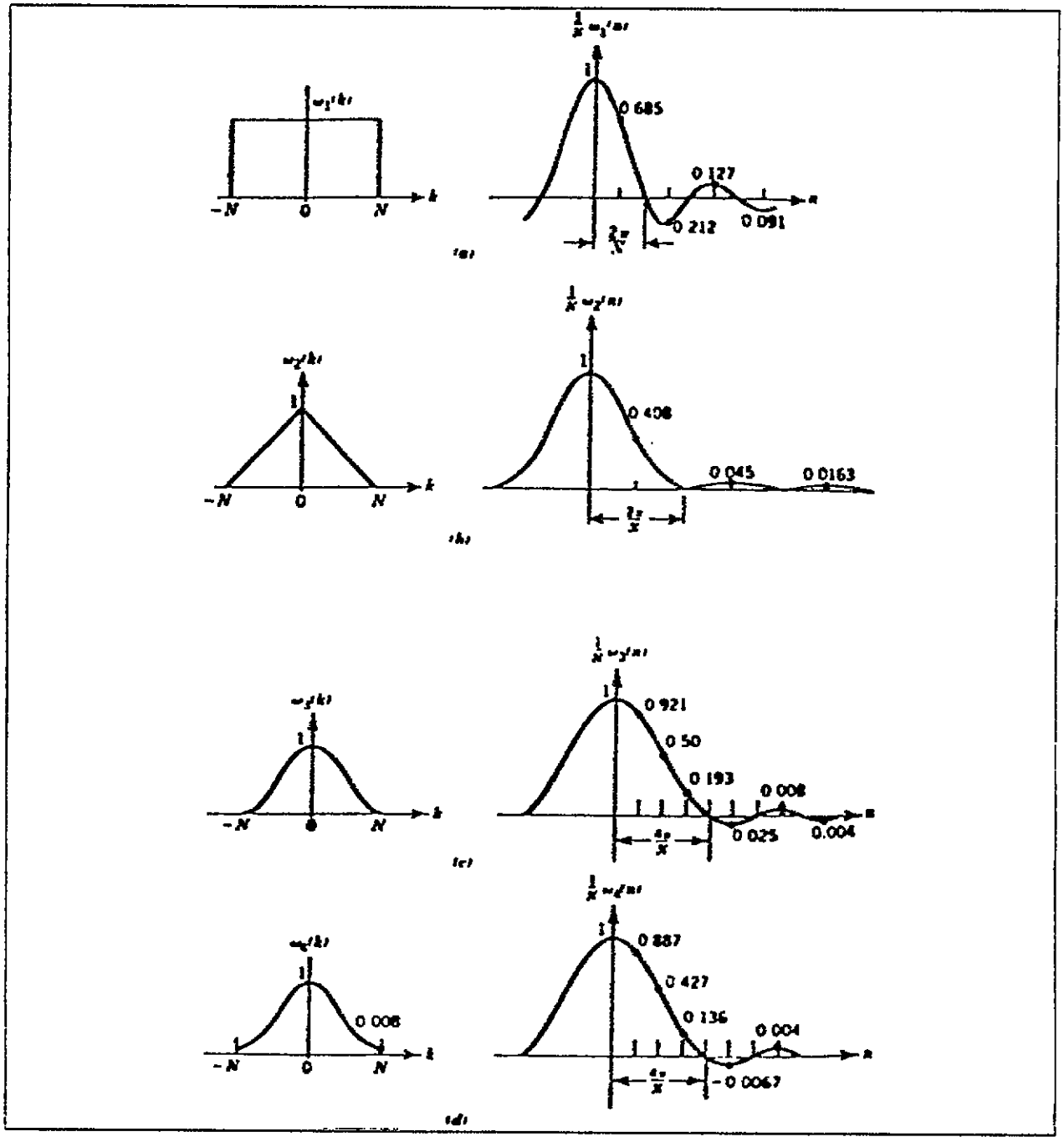

Figure 5-14. Common window functions and their transforms: (a) rectangular, (b) triangular (Fejer-Bartlett), (c) Hanning, (d) Hamming, and (e) Kaiser.

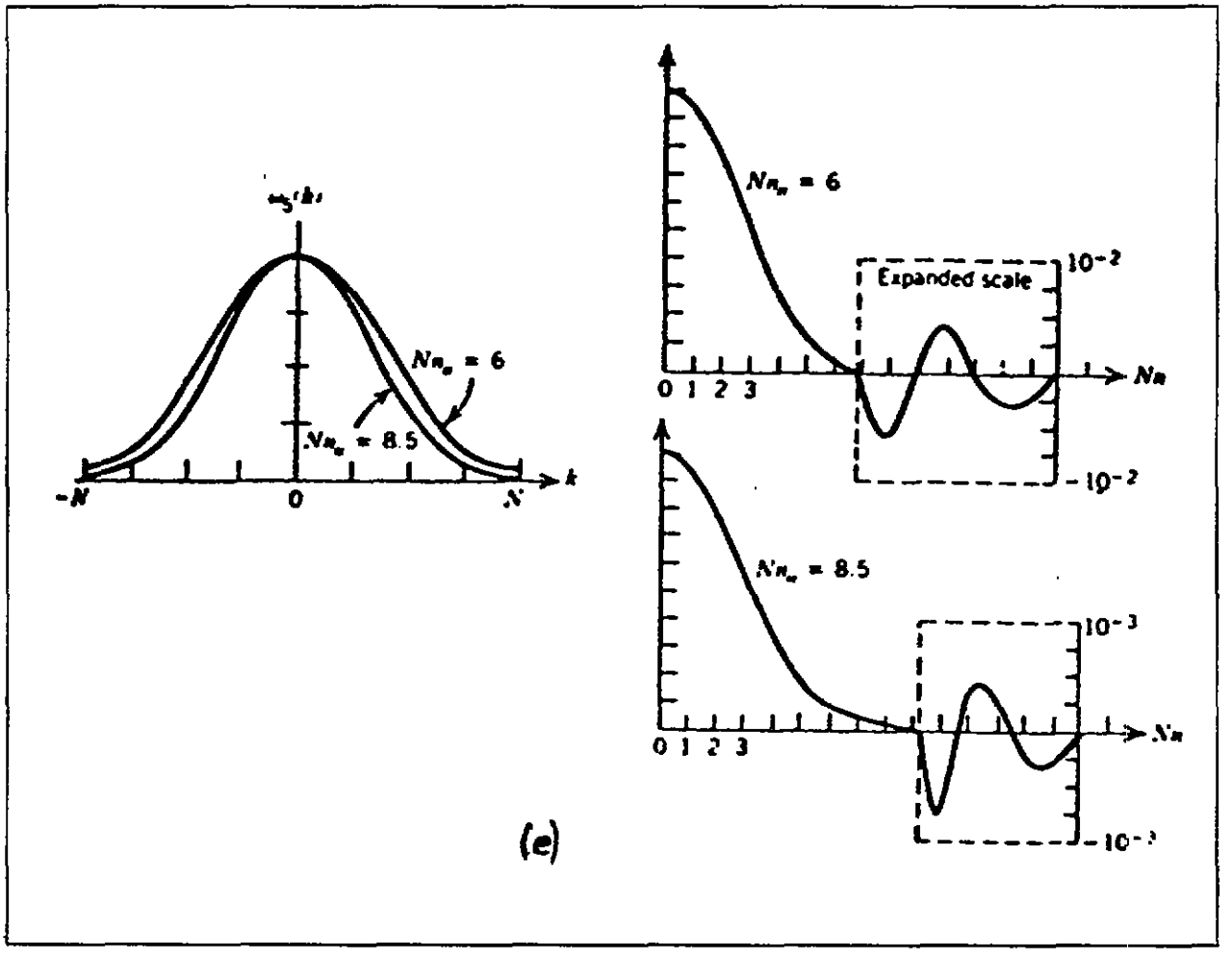

Figure 5-14 (Con.). Common window functions and their transforms: (a) rectangular, (b) triangular (Fejer-Bartlett), (c) Hanning, (d) Hamming, and (e) Kaiser.

Reference 55 compares various windowing functions in one graph, which is inserted here as figure 5-15.

In all of these windowing functions, the side lobes are much smaller than those resulting from the rectangular window, and the main lobes are all wider than those resulting from the rectangular window, producing a much closer approximation to the ideal low-pass filter. The search for the ideal windowing function is a search for the best tradeoff between side lobe amplitude and main lobe width. No FIR filter designed by Fourier transformation and windowing is optimal. The appeal of the technique lies in its simplicity and economy.

Finding the desired transfer function using windows can be summarized as follows:

(1) write the equation for the desired frequency response,

(2) find the impulse response (taking the inverse Fourier transform of the frequency response),

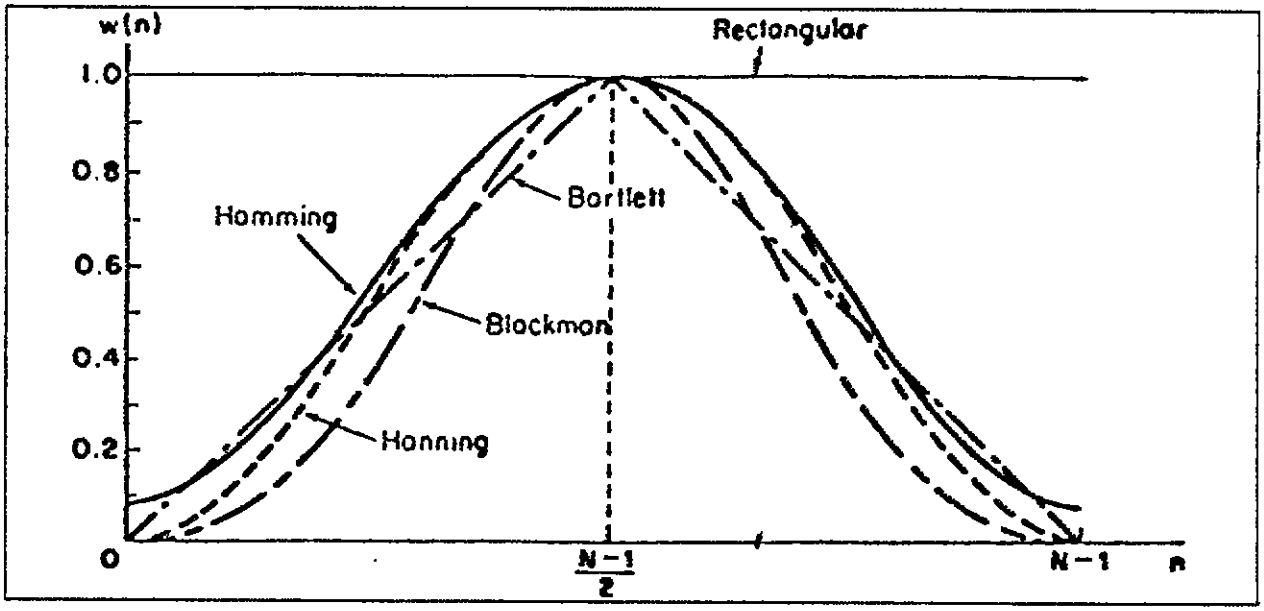

Figure 5-15. Commonly used windows for FIR filter design.

(3) select a windowing function and a window width to meet the required ripple and transition-width specifications, (Window the impulse response accordingly.)

(4) shift the impulse response to make it causal, and

(5) take the Fourier transform of the product of the windowing function and the new impulse response. The result is the desired transfer function.

5.2.2.2 Frequency Sampling Design

The approach of this method is to take samples of the frequency response and to design an FIR transfer function based on these samples. The approach starts by considering the transfer function H(z) of a digital filter, which can be found by taking the z-transform of the impulse response h(n)

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \qquad (5\text{-}46)$$

The impulse response can be found from the frequency response by applying the inverse DFT as

$$h(n) = (1/N) \sum_{k=0}^{N-1} H(k) W^{nk} \qquad (5\text{-}47)$$

where $W = \exp(i2\pi/N)$.

Combining these two relations gives

$$H(z) = (1/N) \sum_{n=0}^{N-1} z^{-n} \sum_{n=0}^{N-1} H(k) W^{nk} \qquad (5\text{-}48)$$

which can be rewritten as

$$H(z) = \frac{1-z^{-N}}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1-z^{-1}W^{k}} \qquad (5\text{-}49)$$

As can be seen, this design procedure consists simply of substituting samples of the desired frequency response into equation (5-49).

As seen from the last equation, there is a pole on the unit circle which leads to marginal stability (see paragraph 3.8.) To ensure stability, multiply $z^{-1}W^{k}$ by a number that is almost one (say $1 - 2^{-12}$). Good accuracy for this method requires many closely spaced samples. This design works particularly well for narrow-band filters in which only a few samples are nonzero. (Reference 55 states that even if more than a few samples are nonzero, the frequency-sampling design method yields excellent results.) A disadvantage of this method is that it lacks flexibility in specifying the passband and stopband cutoff frequencies. In addition, the ripple response for this design procedure is poor. Reference 65 suggests a method in which the ripple response can be greatly improved.

5.2.2.3 Equiripple Design

The approximation error of the frequency sampling design tends to be highest around the transition region and smaller in areas remote from the transition region. The equiripple design affords a way in which the approximation error can be spread out more uniformly.

Of concern are the zero-phase FIR filters with frequency responses of the form

$$H(e^{i\omega}) = \sum_{n=-M}^{M} h(n)e^{-i\omega n} \qquad (5\text{-}50)$$

For zero-phase filters, symmetricalness, h(n) = h(-n), and causality are required. Shift the summands in equation (5-50) are shifted to obtain causality, so that

$$H(e^{i\omega}) = h(0) + \sum_{n=1}^{M} 2h(n)\cos(\omega n) \qquad (5\text{-}51)$$

To be specified for the equiripple curve are the parameters M, δ_1, δ_2, ω_p, and ω_s, where δ_1 and δ_2 are the upper and lower ripple tolerances, and ω_p and ω_s are the passband and stop band cutoff frequencies, (see figure 3-6 or figure 5-16).

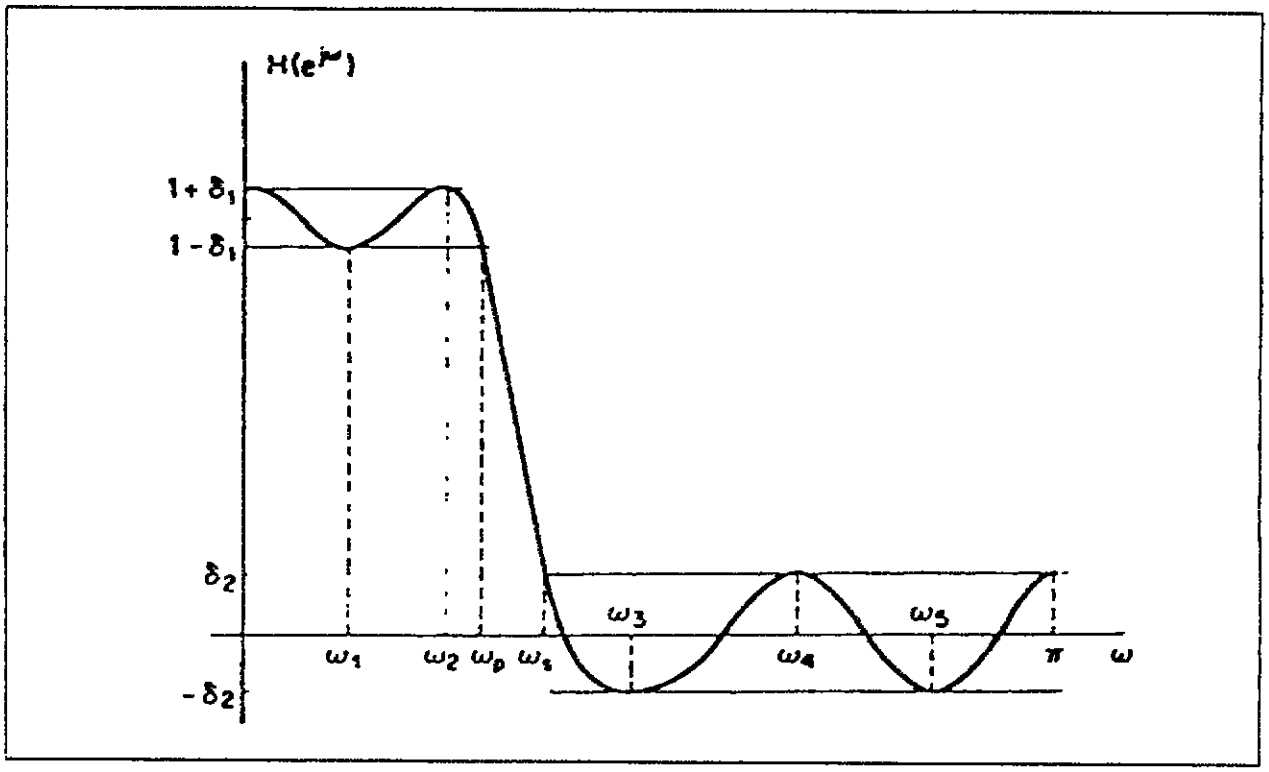

Figure 5-16. Equiripple approximation of a low-pass filter.

Messrs. O. Herrmann and H. W. Schuessler, (references 66 and 67); E. Hofstetter, A. V. Oppenheim, and J. Seigel (references 68 and 69); and J. Seigel (reference 70) developed procedures in which M, δ_1, and δ_2 are held fixed and ω_p and ω_s are solved. Messrs. T. W. Parks and J. H. McClellan, (references 71 and 72) and L. R. Rabiner (references 73 and 74) developed procedures for which M, ω_s, and ω_p are held fixed and for which δ_1 and δ_2 are solved.

In an example using a method by Herrmann and Schuessler, there are five frequencies greater than 0 and less than π where there are maxima and minima in the ripples (see figure 5-16). It can be easily shown that for a symmetric filter of order $N = 2M+1$, there will be at most M+1 local extrema in the interval $0 \leq \omega \leq \pi$.

Consider the fact that at the passband cutoff frequency, the frequency response curve is at the lower tolerance limit about 1 and that at the stop band cutoff frequency, the curve is at the higher tolerance limit about 0. This fact can be used to obtain the following two equations:

$$H(e^{i\omega_p}) = 1-\delta. \tag{5-52}$$

and

$$H(e^{i\omega_s}) = \delta_2. \tag{5-53}$$

By observing either figure 5-16 or equations (5-52) and (5-53), the following set of equations can be written:

$$H(e^{i0}) = 1 + \delta_1, \qquad H(e^{i\pi}) = \delta_2 \tag{5-54}$$

$$H(e^{i\omega_1}) = 1 - \delta_1, \qquad H'(e^{i\omega_1}) = 0 \tag{5-55}$$

$$H(e^{i\omega_2}) = 1 + \delta_1, \qquad H'(e^{i\omega_2}) = 0 \tag{5-56}$$

$$H(e^{i\omega_3}) = -\delta_2, \qquad H'(e^{i\omega_3}) = 0 \tag{5-57}$$

$$H(e^{i\omega_4}) = \delta_2, \qquad H'(e^{i\omega_4}) = 0 \tag{5-58}$$

$$H(e^{i\omega_5}) = -\delta_2, \qquad H'(e^{i\omega_5}) = 0 \tag{5-59}$$

For this method, there is the flexibility to decide which of the M-1 frequencies properly between 0 and π should be in the pass-band and which ones should be in the stop band. In this case, M+1=7, there are seven unknown coefficients (h(n)) in equation (5-51). There are 5 unknown frequencies $\omega_1,\ldots,\omega_5$ at which extrema occur, so there are 12 equations in 12 unknowns. These equations are nonlinear and must be solved by an iterative procedure.

Generally, there are 2M equations in 2M unknowns. This approach has been found to be satisfactory for orders of M = 30 or lower, and it provides the narrowest transition between passband and stop band.

CHAPTER 6

KALMAN FILTERS

A Kalman filter is a linear, recursive algorithm for computing an optimal estimate from measurements, some of which may contain noise. The noise on the measurements is assumed to be white; namely, the noise values are not correlated over time. Also assumed is knowledge of the statistics of the noise on the measurements. The algorithm is recursive and thus requires an initial estimate to start the filter as well as a guess as to the correctness of that estimate. Finally, the algorithm is linear, consisting of matrix equations. (Reference 3 has an excellent intuitive introduction to Kalman filtering in its first chapter.)

The Kalman filter is most often used as a data processing algorithm (a computer program) and can be extremely efficient, requiring a minimum of computer storage and using all available data by weighing the data measurements. The filter has the ability to take several different types of data and generate an estimate of a totally different quantity. Because it is a predictor-corrector, it can generate its next estimate based on less current data than would be needed for a directly calculated solution. The data need not be entered at equally spaced time intervals, at the same time, or in certain sequences. The filter allows the user to apply the knowledge of the behavior and statistics of both the measurements and the quantities to be estimated to obtain the solution; in fact, these models and statistics may vary with time. In addition, a self-contained error analysis is included in its equations. Finally, as a predictor, it is useful for real-time control.

A few caveats apply, however. To begin with, there is no "general" Kalman filter. Each algorithm is dependent on the quantities to be estimated and their dynamics, the measurements available and their statistics, and the initial values needed to start the algorithm. Secondly, the algorithm is most efficient if matrix inversions can be avoided. Matrix inversions can usually be done, although it depends on the application, that is, the particular Kalman filter written. Thirdly, a Kalman filter will update with less current data than is required for a calculated solution, but if updating with this data is done for too long, the estimate may become grossly invalid. The filter is said to have diverged. Fourthly, the order of the input of data is unimportant only for the basic Kalman filter: a filter whose dynamic and measurement models are both linear. Though all derivations and claims of optimality are valid only for such filters, linearity is rare in the real world. Hence, suboptimality is often settled for using an extended Kalman filter. Sometimes it is found that permuting the order in which data is

entered may significantly change the estimate. Finally, the validity of the self-contained error analysis is dependent on the degree to which the noise statistics fit the theoretical assumptions. For example, biases in the data will not be indicated by the output error covariance matrix.

Before describing a Kalman filter, the description of some terms are in order.

The "expected value" of g(x) of a random variable X whose frequency function is f(x) is

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

Loosely speaking, the expected value of g(x) is the average of g(x).

An "estimate" is a computed value of a quantity. For example, the sample mean

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{\eta}$$

is an estimate of true population mean μ.

A statistic t is called an "unbiased estimate" or "unbiased estimator" of the parameter ν if $E[t] = \nu$. For example, the sample mean x can be shown to be an unbiased estimate of the true population mean μ.

An "optimal estimate" is one that minimizes the variance of an estimate and is unbiased. When using an optimal filter such as a filter computing an optimal estimate, it is assumed that the exact descriptions of the system dynamics and the measurement process are known. In addition, an optimal filter must model <u>all</u> error sources in the system including unmodeled parameters, linearization errors, leaky attitude controls, and solar winds.

Error sources mentioned in the previous paragraph constitute "process noise," noise that stems from mismodeling. Other error sources include those of "measurement noise," which stems from faulty measuring devices or the misuse of those devices. (Because the Kalman filter is based on least squares, an error source inherent in the Kalman filter is the difference between the estimate sample measurement values and the true population measurement values.) An example dealing with unmodeled parameters is filtering a position parameter without including the

velocity or acceleration parameter into the model, which where including these parameters could facilitate filtering the position parameter.

Leaky attitude controls and solar winds are factors that can lead a spacecraft off course. They are usually considered negligible and as such are normally not included in the models. The errors that may result from their not being included constitute process noise.

A suboptimal filter is a filter that does not take into account all the factors that have been mentioned in describing an optimal filter. A "predictor-corrector" is a recursive algorithm that has two steps to each recursion: a predictor and a corrective step. A predictor step is the state of the system predicted by using the output data of the preceding recursions in which the nth state of the system is predicted using the first n-1 states. A corrective step is the given input nth state of the system used to correct the predicted nth state of the system, thus producing the final output nth state of the system. (The nth state of the system is the state of the system at the nth recursion.)

6.1 Linear Discrete Kalman Filter

In the case of a linear discrete Kalman filter, the dynamics of the quantities to be estimated may be described by linear difference equations. Additionally, the relationship between the measurements, taken at discrete times, and the estimated quantities is linear.

6.1.1 Definitions

To write such a Kalman filter, begin with these definitions.

(1) A model describing the dynamics of the quantities to be estimated. This model will be of the form

$$\underline{x}(k) = \Phi(k,k-1)\underline{x}(k-1) + G(k,k-1)\underline{w}(k-1) \qquad (6\text{-}1)$$

where

k refers to the kth time point t_k;

$\underline{x}(k)$ is the set of quantities to be estimated, arranged in a vector, and called the state vector;

$\Phi(k,k-1)$ is called the transition matrix and describes the change in the state vector from time t_{k-1} to t_k;

$G(k,k-1)$ is called the input matrix; for practical purposes, it is generally taken to be equal to the identity matrix;

and

$\underline{w}(k-1)$ is called the plant or process noise. The vector $\underline{w}(k-1)$ is assumed to contain zero-mean white noise with Gaussian distribution. The vector resulting from the product $G(k,k-1)\underline{w}(k-1)$ represents the unknown portion of the dynamic model. The covariance matrix of $\underline{w}(k)$ is designated by $Q(k)$.

(2) An equation relating the state vector to the measurements. The form of this equation is

$$\underline{y}(k) = H(k)\underline{x}(k) + \underline{v}(k) \qquad (6\text{-}2)$$

where

$\underline{y}(k)$ is the measurement at time t_k;

$H(k)$ is called the measurement matrix and linearly relates the measurements to the state variables;

and

$\underline{v}(k)$ is the measurement noise. The vector $\underline{v}(k)$ is assumed to contain zero-mean white noise with Gaussian distribution. The covariance matrix of $\underline{v}(k)$ is designated by $R(k)$.

(3) Finally, the following quantities are needed:

$\hat{\underline{x}}(0)$ the initial state estimate; and

$P(0)$ the corresponding error covariance matrix. (In the Kalman filter algorithm, P_k is the covariance matrix of the estimate error of the kth state vector. That is, $P_k = E[(\underline{x}(k)-\hat{\underline{x}}_k)(\underline{x}(k)-\hat{\underline{x}}_k)^T]$.)

To write a Kalman filter, three matrices, I, G, and H are needed to define the dynamic and measurement structures as well as the matrices, Q and R, to define the dynamic and measurement statistics. Also needed are the initial values, $\hat{\underline{x}}(0)$ and $P(0)$.

Given these items, the filter is written using the equations described next. In these equations, the notation $\hat{\underline{x}}$ denotes the estimate of the state vector $\underline{x}$. The notation $\hat{\underline{x}}(a/b)$ signifies the estimate of the state vector at time t_a given measurements taken up to and including time t_b.

6.1.2 Kalman Filter Algorithm

The first step in this Kalman filter algorithm is the extrapolation or time update for the estimate and its covariance matrix. These equations are

$$\hat{\underline{x}}(k/k-1) = \Phi(k,k-1)\hat{\underline{x}}(k-1/k-1) \tag{6-3}$$

and

$$P(k/k-1) = \Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1) + G(k,k-1)Q(k-1)G^T(k,k-1). \tag{6-4}$$

The next step is to compute the weighing matrix K, called the Kalman gain, by

$$K(k) = P(k/k-1)H^T(k)[H(k)P(k/k-1)H^T(k)+R(k)]^{-1} \tag{6-5}$$

Finally, the output is obtained by

$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + K(k)[\underline{y}(k) - H(k)\hat{\underline{x}}(k/k-1)] \tag{6-6}$$

and

$$P(k/k) = [I - K(k)H(k)]\ P(k/k-1), \tag{6-7}$$

where I is the proper size identity matrix.

$\hat{\underline{x}}(k/k)$ in equation (6-6) is, ultimately, the result for the kth step in the algorithm. P(k/k) of equation (6-7) is used as P(k-1/k-1) in equation (6-4) in a recursive step. Equations (6-3) through (6-7) represent one recursive step in the Kalman filter algorithm.

Table 6-1 is obtained from reference 2 and gives a summary of the entire algorithm. The top two boxes give the original model equations, initial conditions, and assumptions. Equations (6-1) and (6-2) are equivalent to the system and measurement model equations. (The $\underline{z}$ in the table is the same as the $\underline{y}$ vector of equation (6-2).) The "other assumptions" equation in the second box means that the system model errors and the measurement noises are uncorrelated. The equations in the bottom two boxes correspond to equations (6-3) through (6-7), which comprise one recursive step in the algorithm.

TABLE 6-1. SUMMARY OF DISCRETE KALMAN FILTER EQUATIONS

System Model	$\underline{x}_k = \Phi_{k-1}\underline{x}_{k-1} + \underline{w}_{k-1}, \quad \underline{w}_k \sim N(\underline{0}, Q_k)$
Measurement Model	$\underline{z}_k = H_k\underline{x}_k + \underline{v}_k, \quad \underline{v}_k \sim N(\underline{0}, R_k)$
Initial Conditions	$E[\underline{x}(0)] = \hat{\underline{x}}_0, E[(\underline{x}(0) - \hat{\underline{x}}_0)(\underline{x}(0) - \hat{\underline{x}}_0)^T] = P_0$
Other Assumptions	$E[\underline{w}_k\underline{v}_j^T] = 0$ for all j, k
State Estimate Extrapolation	$\hat{\underline{x}}_k(-) = \Phi_{k-1}\hat{\underline{x}}_{k-1}(+)$
Error Covariance Extrapolation	$P_k(-) = \Phi_{k-1}P_{k-1}(+)\Phi_{k-1}^T + Q_{k-1}$
State Estimate Update	$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k[\underline{z}_k - H_k\hat{\underline{x}}_k(-)]$
Error Covariance Update	$P_k(+) = [I - K_kH_k]P_k(-)$
Kalman Gain Matrix	$K_k = P_k(-)H_k^T[H_kP_k(-)H_k^T + R_k]^{-1}$

(-) and (+) are used to denote the times immediately before and immediately after a discrete measurement. For example, $P_k(+)$ and $P_k(-)$ in the table are equivalent to P(k/k-1) in equation (6-7).

Several quantities in equations (6-3) through (6-7) are of particular interest. The vector $H(k)\hat{\underline{x}}(k/k-1)$ in equation (6-6) is called the predicted measurement and its difference with the actual measurement, given by

$$\underline{y}(k) - H(k)\underline{x}(k/k-1) \tag{6-8}$$

is called the innovation. This vector is often used for testing for filter divergence or for data editing.

The matrix

$$H(k)P(k/k-1)H^T(k) + R(k) \tag{6-9}$$

in equation (6-5) is considered the covariance matrix of the innovations and is usually chosen to be one-dimensional, namely, a scalar, in the following manner. If only one measurement is input at time t_k, then in equation (6-2) $\underline{y}(k)$ is a scalar, H(k) has only one row, and $\underline{v}(k)$ is a scalar. The matrix (6-9) is then a scalar, whatever the size of P(k/k-1). If more than one

measurement is available at time t_k, it is usually more economical in computer time to process each measurement individually, since matrix inversions are avoided. Hence, the first measurement used would be preceded by a time update (equations (6-3) and (6-4). (The estimate $\hat{\underline{x}}$ and the estimate error covariance matrix P being is updated from time t_{k-1} to time t_k. (equations (6-3) and (6-4)) The other measurements at time t_k would not need a time update, so only equations (6-5), (6-6), and (6-7) would be used with the old $\hat{\underline{x}}(k/k-1)$ and $P(k/k-1)$ of the previous step being replaced in each equation by the new ones.

As mentioned before, for a Kalman filter with linear models and the assumed noise statistics, the order in which all the measurements at time t_k are processed is not significant. Most importantly, processing the measurements individually will give the same output as processing them together in a measurement vector if there is no cross correlation among the measurements at time t_k. Usually this is the case or at least can be assumed true with a minimum of error.

The preceding equations will easily yield a computer program which is a working Kalman filter. The problem remains, however, to make sure the filter does the job for which it was intended. Once written, the filter must be subjected to extensive simulation testing. Testing is necessary to detect modeling errors, statistical assumption errors, inappropriate initial conditions, biases, correlated noise sequences, and finally to determine tuning value.

Filter tuning is the process of achieving the best possible estimation performance from a filter once its structural form has been specified. In a Kalman filter, the structural form is specified by the matrices Φ, G, and H. The initial estimate error covariance matrix P(0), the model error covariance matrix Q, and the measurement noise covariance matrix R are the variables modified during tuning. The use of Q and R is normally based on the knowledge of $\underline{w}$ (the system model error) and $\underline{v}$ (the measurement noise). These vectors, $\underline{w}$ and $\underline{v}$, account for actual noises and disturbances in the physical system as well as inadequacies in the dynamic and measurement models.

6.2 The Linear Continuous Kalman Filter

For a linear continuous Kalman filter, start with the dynamic model

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + G(t)\underline{w}(t) \qquad (6\text{-}10)$$

and the measurement model

$$\underline{y}(t) = H(t)\underline{x}(t) + \underline{v}(t) \qquad (6\text{-}11)$$

with initial values $\underline{x}(0)$ and $P(0)$, where $\underline{w}(t)$ and $\underline{v}(t)$ are zero-mean white noise processes uncorrelated with $\underline{x}(0)$ and have covariance matrices $Q(t)$ and $R(t)$ (see table 6-2).

TABLE 6-2. SUMMARY OF CONTINUOUS KALMAN FILTER EQUATIONS (WHITE MEASUREMENT NOISE)

System Model	$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + G(t)\underline{w}(t), \quad \underline{w}(t) \sim N(\underline{0}, Q(t))$
Measurement Model	$\underline{y}(t) = H(t)\underline{x}(t) + \underline{v}(t), \quad \underline{v}(t) \sim N(\underline{0}, R(t))$
Initial Conditions	$E[\underline{x}(0)] = \hat{\underline{x}}_0, \; E[(\underline{x}(0) - \hat{\underline{x}}_0)(\underline{x}(0) - \hat{\underline{x}}_0)^T] = P_0$
Other Assumptions	$R^{-1}(t)$ exists
State Estimate	$\dot{\hat{\underline{x}}}(t) = F(t)\hat{\underline{x}}(t) + K(t)[\underline{y}(t) - H(t)\hat{\underline{x}}(t)], \; \hat{\underline{x}}(0) = \hat{\underline{x}}_0$
Error Covariance Propagation	$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - K(t)R(t)K^T(t), \; P(0) = P_0$
Kalman Gain Matrix	$K(t) = P(t)H^T(t)R^{-1}(t)$ when $E[\underline{w}(t)\underline{v}^T(\tau)] = 0$ $= [P(t)H^T(t) + G(t)C(t)]R^{-1}(t)$ when $E[\underline{w}(t)\underline{v}^T(\tau)] = C(t)\delta(t-\tau)$

Then the filter equations are

$$\dot{\hat{\underline{x}}}(t) = F(t)\hat{\underline{x}}(t) + K(t)[\underline{y}(t) - H(t)\hat{\underline{x}}(t)], \qquad (6\text{-}12)$$

and

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - K(t)R(t)K^T(t) \qquad (6\text{-}13)$$

with

$$K(t) = [P(t)H^T(t) + G(t)C(t)]\,R^{-1}(t) \qquad (6\text{-}14)$$

where $E\{\underline{w}(t)\underline{v}^T(s)\} = C(t)\delta(t-s)$, with E representing the expected value and δ, the Dirac delta function. The three equations comprise one recursive step in the continuous Kalman filter algorithm.

The problem is, in part, to solve P(t) in equation (6-13), which is known as the "matrix Riccati equation." (Reference 2 gives methods for solving this equation.) Then plug P(t) into equation (6-14) to solve for K(t), which is used in equation (6-12) to obtain the final desired result $\hat{\underline{x}}(t)$ for a particular recursive step. (See reference 2 for further discussion.)

6.3 Extended Kalman Filters

An extended Kalman filter is a data processing algorithm which is based on the Kalman filter algorithm and which efficiently provides estimates for nonlinear problems. Such filters are suboptimal because no theoretical optimality of the estimate can be proven. Either or both of the given models (continuous or discrete, otherwise known as "dynamic" or "measurement") of a Kalman filter may be nonlinear for an extended Kalman filter to be required. The extended Kalman filter is still a linear algorithm, however.

For an extended Kalman filter, assume the dynamic model is

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t),t) + G(\underline{x}(t),t)\underline{w}(t) \qquad (6\text{-}15)$$

with initial conditions $\underline{x}(t_0)$ and $P(0) = \text{cov}\{\underline{x}(t_0),\underline{x}(t_0)\}$, where $\underline{w}(t)$ is a zero-mean, Gaussian noise vector, such that $\text{cov}\{\underline{w}(t),\underline{w}(s)\} = Q(t)\delta(t-s)$, and $\text{cov}\{\underline{x}(t_0),\underline{w}(t)\} = 0$ for $t \geq t_0$. The notation cov{.,.} represents the covariance matrix of the two vectors. The δ is the Dirac delta function. The vector $\underline{f}$ is a nonlinear function of the state vector $\underline{x}(t)$ and of the time t. Notice that the model (6-15) is not completely general since the dynamic noise is assumed additive.

The measurement model is given by

$$\underline{y}(t_k) = \underline{h}(\underline{x}(t_k),t_k) + \underline{v}(t_k),\ k = 0,1,\ldots, \qquad (6\text{-}16)$$

where $\underline{v}(t_k)$ is the zero-mean measurement noise vector, with $\text{cov}\{\underline{v}(t_k),\underline{v}(t_k)\} = R(k)$, $\text{cov}\{\underline{v}(t_k),\underline{v}(t_j)\} = 0$, $\text{cov}\{\underline{x}(t_0),\underline{v}(t_k)\} = 0$, and $\text{cov}\{\underline{w}(t),\underline{v}(t_k)\} = 0$.

Note that the measurement model, though nonlinear, is still discrete since the measurements are taken at discrete times. Also assume the initial values $\hat{\underline{x}}(0)$ and P(0) are given.

Then the extended Kalman filter update from the measurement at time t_k to the measurement at time t_{k+1} is accomplished in the following manner:

(1) The state vector update is achieved by using the vector function $\underline{f}$ of equation (6-15). Thus, the equation

$$\dot{\hat{\underline{x}}}(t/t_k) = \underline{f}(\hat{\underline{x}}(t/t_k),t) \tag{6-17}$$

is integrated for $t_k \leq t \leq t_{k+1}$, resulting in $\hat{\underline{x}}(t/t_k)$.

(2) The matrix

$$F(\hat{\underline{x}}(t),t) = \left.\frac{\delta \underline{f}(\underline{x}(t),t)}{\delta \underline{x}}\right|_{\underline{x}(t) = \hat{\underline{x}}(t/t_k)} \tag{6-18}$$

is calculated.

For the rest of the chapter, $F(\hat{\underline{x}}(t),t)$ will be denoted $F(t)$, $\underline{f}(\underline{x}(t),t)$ will be denoted $\underline{f}$, and $\underline{x}(t)$ will be denoted $\underline{x}$ for the sake of convenience.

(3) The transition matrix $\Phi(t_{k+1},t_k)$ is calculated using

$$\frac{\partial \Phi(t,t_k)}{\partial t} = F(t)\Phi(t,t_k) \tag{6-19}$$

where $t_k \leq t \leq t_{k+1}$ and $\Phi(t_k,t_k)$ is the identity matrix.

(4) The state noise covariance matrix Q* is calculated using

$$Q^*(t_{k+1}) = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1},t)G(t)Q(t)G^T(t)\Phi^T(t_{k+1},t)dt. \tag{6-20}$$

(5) The measurement matrix $H(t_{k+1})$ is calculated by

$$H(t_{k+1}) = \left.\frac{\partial \underline{h}}{\partial \underline{x}}\right|_{\underline{x} = \hat{x}(t_{k+1}/t_k),} \tag{6-21}$$

where $\hat{\underline{x}}(t_{k+1}/t_k)$ was obtained in step (1).

The update is not ready to be performed. To obtain the gain matrix $K(t_{k+1})$, first extrapolate the state covariance matrix using equation

$$P(t_{k+1}/t_k) = \Phi(t_{k+1},t_k)P(t_k/t_k)\Phi^T(t_{k+1},t_k) + Q^*(t_{k+1}). \tag{6-22}$$

Then

$$K(t_{k+1}) = P(t_{k+1}/t_k)H^T(t_{k+1})[H(t_{k+1})P(t_{k+1}/t_k)H^T(t_{k+1}) + R(t_{k+1})] \tag{6-23}$$

The state update is then

$$\hat{x}(t_{k+1}/t_{k+1}) = \underline{\hat{x}}(t_{k+1}/t_k) + K(t_{k+1})[\underline{y}(t_{k+1}) - \underline{h}(\underline{\hat{x}}(t_{k+1}/t_k), t_k)]. \tag{6-24}$$

Equation (6-24) gives the desired update at time t_{k+1}. To complete the filter update, compute the state covariance matrix at time t_{k+1} by

$$P(t_{k+1}/t_{k+1}) = [I - K(t_{k+1})H(t_{k+1})]P(t_{k+1}/t_k), \tag{6-25}$$

where I is the appropriately sized identity matrix.

Further modifications of the basic Kalman filter equations are possible and often necessary for the proper functioning of a Kalman filter. If the noise vectors are not zero mean, the state vector may be expanded to solve for bias estimates. If the noise sequences are not white, shaping filters may be used. If the noise statistics are unknown, adaptive estimation schemes may be added to determine these statistics. If computer word length affects the numerical stability of the filter, square root type filters can be written. The literature is filled with such Kalman filter applications and more are being added every day as Kalman filters are used to solve harder and harder problems.

A disadvantage of the extended Kalman filter is that the Kalman gain $K(t_{k+1})$ and the error covariance update matrix $P(t_{k+1}/t_{k+1})$ must be computed in real time. They cannot be precomputed before the measurements are collected and stored in computer memory as can be done when using the basic Kalman filter because $K(t_{k+1})$ and $P(t_{k+1}/t_{k+1})$ are both dependent on $\underline{\hat{x}}(t_{k+1}/t_k)$. In equations (6-23) and (6-25), it is seen that $H(t_{k+1})$ is needed to calculate $K(t_{k+1})$ and $P(t_{k+1}/t_{k+1})$. $H(t_{k+1})$, given in equation (6-21), is dependent on $\underline{\hat{x}}(t_{k+1}/t_k)$. $H(t_{k+1})$ is actually shorthand for $H(\underline{\hat{x}}(t_{k+1}/t_k))$. Needed is the estimate update, and hence the measurements themselves, to calculate the Kalman gain and the error covariance update.

After the extended Kalman filter is designed, a "sensitivity analysis" is sometimes performed. A sensitivity analysis comprises a set of analyses to determine the sensitivity of the filter design to any possible differences between this suboptimal filter and a filter that fits the optimal mold exactly.

References 1 and 2 discuss solutions to several of the problems mentioned in the preceding paragraphs. Since the discussion here was limited to conventional real time Kalman filters, the reader may also wish to consult references 1 and 2 for a state-of-the-art exposition on post-flight Kalman smoothers.

6.4 Example

A simple example will illustrate an extended Kalman filter as described in paragraph 6.3. A moving target is tracked by N stations, each measuring at times t_k, k=1,2,..., m, the distance of the target from the station. Estimate the target's position and velocity at time t_j, $1 \le j \le m$, using all the range measurements up to and including those at time t_j.

In this example, the state vector $\underline{x}(t)$ is chosen to be the six-vector consisting of the position and velocity of the target in a geocentric coordinate system, so

$$\underline{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ vx(t) \\ vy(t) \\ vz(t) \end{bmatrix} \tag{6-26}$$

The continuous dynamic model corresponding to equation (6-15) is

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t)) + G(\underline{x}(t))\underline{w}(t) \tag{6-27}$$

where

$$\underline{f}(\underline{x}(t)) = \begin{bmatrix} vx(t) \\ vy(t) \\ vz(t) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad G(\underline{x}(t)) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6-28}$$

and

$$\underline{w}(t) = \begin{bmatrix} ax(t) \\ ay(t) \\ az(t) \end{bmatrix} \tag{6-29}$$

where ax, ay, and az are Gaussian white noise errors. They can also be considered geocentric components of target acceleration. Of course, $\underline{f}(\underline{x}(t))$, G, and $\underline{w}(t)$ could have been assigned other values to satisfy equation (6-27). In real-life situations, however, the values assigned to $\underline{f}$, G, and $\underline{w}$ are as given above.

The initial conditions $\hat{\underline{x}}(t_0)$ and P(0) are assumed to be provided before time t_1. A simple (and usually inadequate) method is to take three ranges prior to time t_1 and triangulate to get $\hat{x}(t_0)$, $\hat{y}(t_0)$, and $\hat{z}(t_0)$ and set $\hat{vx}(t_0) = \hat{vy}(t_0) = \hat{vz}(t_0) = 0$. The 6x6 matrix P(0) can be chosen to be diagonal with values on the main diagonal to reflect the filter designer's confidence or lack thereof in the method of determining $\hat{\underline{x}}(t_0)$. (Generally, the greater the confidence, the larger the values.)

The matrix Q(t) where cov $\{\underline{w}(t),\underline{w}(s)\} = Q(t)\delta(t-s)$ is assumed to be $\sigma^2 I_3$, where I_3 is the 3x3 identity matrix and σ^2 is chosen to compensate for unmodeled accelerations. The value of σ may be determined by subsequent tuning studies.

The measurement model is given in equation (6-16) as

$$\underline{y}(t_k) = \underline{h}(\underline{x}(t_k),t_k) + \underline{v}(t_k).$$

For this example, the vector $\underline{y}(t_k)$ is N-dimensional consisting of the N ranges available at time t_k. For any i, $1 \le i \le N$, the ith component of $\underline{h}(\underline{x}(t_k),t_k)$ is

$$\sqrt{(x(t_k) - X_i)^2 + (y(t_k) - Y_i)^2 + (z(t_k) - Z_i)^2} \qquad (6\text{-}30)$$

where $x(t_k)$, $y(t_k)$, and $z(t_k)$ are components of the state vector at time t_k and (X_i, Y_i, Z_i) is the location of the ith tracking station in geocentric coordinates. The matrix R(k) = cov $\{\underline{v}(t_k),\underline{v}(t_k)\}$ is assumed to be diagonal with main diagonal element i, $1 \le i \le N$, equal to the expected variance of the measurements provided by the ith ranging station.

The measurements in this example are going to be processed individually as described in paragraph 6.1. Hence, for each t_k, there are N measurements to process which satisfy the scalar equation

$$y_i(t_k) = h_i(\underline{x}(t_k)) + v_i(t_k), \quad i = 1,2,\ldots, N, \qquad (6\text{-}31)$$

where

$$h_i(\underline{x}(t_k)) = \sqrt{(x(t_k) - X_i)^2 + (y(t_k) - Y_i)^2 + (z(t_k) - Z_i)^2}. \qquad (6\text{-}32)$$

The matrix $R_i(k) = \text{cov}\{\epsilon_i(t_k), v_i(t_k)$ is then a scalar.

Now proceed with the steps described in paragraph 6.3. These steps describe the quantities needed to estimate the position and velocity of the target at time t_{k+1}, given the estimate at time t_k, and the measurements at time t_{k+1}.

(1) The equation

$$\dot{\underline{\hat{x}}}(t/t_k) = \underline{f}(\underline{\hat{x}}(t/t_k)) \tag{6-33}$$

must be integrated for $t_k \leq t \leq t_{k+1}$ to get $\underline{\hat{x}}(t/t_k)$. According to equation (6-27), equation (6-33) is written out as

$$\begin{bmatrix} \hat{x}(t/t_k) \\ \hat{y}(t/t_k) \\ \hat{z}(t/t_k) \\ \hat{vx}(t/t_k) \\ \hat{vy}(t/t_k) \\ \hat{vz}(t/t_k) \end{bmatrix} \qquad \begin{bmatrix} \hat{vx}(t/t_k) \\ \hat{vy}(t/t_k) \\ \hat{vz}(t/t_k) \\ 0 \\ 0 \\ 0 \end{bmatrix} . \tag{6-34}$$

To integrate the vector equation (6-34), this example begins with the lower three components of the vector. The fourth component of the vector equation (6-34) is

$$\dot{\hat{vx}}(t/t_k) = 0. \tag{6-35}$$

Integrate equation (6-35) for $t_k \leq t \leq t_{k+1}$ to obtain

$$\hat{vx}(t_{k+1}/t_k) = \hat{vx}(t_k/t_k). \tag{6-36}$$

Similar computations hold for $\hat{vy}$ and $\hat{vz}$. Since the velocity estimate is constant over the interval $[t_k, t_{k+1}]$, the first component of equation (6-34) can be rewritten as

$$\dot{\hat{x}}(t/t_k) = vx\ (t_k/t_k). \tag{6-37}$$

Integrate both sides to obtain

$$\int_{t_k}^{t_{k+1}} \dot{\hat{x}}(t/t_k)\ dt = \hat{vx}(t_k/t_k) \int_{t_k}^{t_{k+1}} dt \tag{6-38}$$

so that

$$\hat{x}(t_{k+1}/t_k) - \hat{x}(t_k/t_k) = \hat{vx}(t_k/t_k)(t_{k+1} - t_k) \tag{6-39}$$

or

$$\hat{x}(t_{k+1}/t_k) = \hat{x}(t_k/t_k) + \hat{vx}(t_k/t_k)(t_{k+1} - t_k). \quad (6\text{-}40)$$

Similar computations yield $\hat{y}(t_{k+1}/t_k)$ and $\hat{z}(t_{k+1}/t_k)$. (6-41)

Thus, the state vector time update equation is

$$\begin{bmatrix} \hat{x}(t_{k+1}/t_k) \\ \hat{y}(t_{k+1}/t_k) \\ \hat{z}(t_{k+1}/t_k) \\ \hat{vx}(t_{k+1}/t_k) \\ \hat{vy}(t_{k+1}/t_k) \\ \hat{vz}(t_{k+1}/t_k) \end{bmatrix} = \begin{bmatrix} \hat{x}(t_k/t_k) + \hat{vx}(t_k/t_k)(t_{k+1} - t_k) \\ \hat{y}(t_k/t_k) + \hat{vy}(t_k/t_k)(t_{k+1} - t_k) \\ \hat{z}(t_k/t_k) + \hat{vz}(t_k/t_k)(t_{k+1} - t_k) \\ \hat{vx}(t_k/t_k) \\ \hat{vy}(t_k/t_k) \\ vz(t_k/t_k) \end{bmatrix} \quad (6\text{-}42)$$

(2) The next step is to compute the matrix F(t) of equation (6-18). Using equation (6-27), the i,jth element of the matrix F(t) is

$$\left[\frac{\partial \underline{f}}{\partial \underline{x}}\right]_{ij} = \frac{\partial (\underline{f})_i}{\partial (\underline{x})_j}, \qquad 1 \le i,j \le 6, \quad (6\text{-}43)$$

evaluated at $\underline{x} = \hat{\underline{x}}(t/t_k)$. For $1 \le i \le 3$ and $1 \le j \le 3$,

$$\frac{\partial (\underline{f})_i}{\partial (\underline{x})_j} \text{ is of the form } \frac{\partial vx(t)}{\partial x(t)}, \quad (6\text{-}44)$$

which is zero, since velocity is not dependent on position.

For $1 \le i \le 3$ and $4 \le j \le 6$, if $i \neq j$.

$$\frac{\partial (\underline{f})_i}{\partial (\underline{x})_j} \text{ is of the form } \frac{\partial vx(t)}{\partial vy(t)}, \quad (6\text{-}45)$$

which is zero. If i=j,

$$\frac{\partial (\underline{f})_i}{\partial (\underline{x})_j} \text{ is of the form } \frac{\partial vx(t)}{\partial vx(t)}, \quad (6\text{-}46)$$

which is 1. For $4 \leq i \leq 6$ and all j,

$$\frac{\partial (\underline{f})_i}{\partial (\underline{x})_j} = 0 \qquad (6\text{-}47)$$

Hence,

$$F(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \qquad (6\text{-}48)$$

(3) The next step is to compute the transition matrix Φ according to the matrix differential equation (6-19) which is

$$\frac{\partial\ \Phi(t,t_k)}{\partial t} = F(t)\Phi(t,t_k)$$

where $t_k \leq t \leq t_{k+1}$ and $\Phi(t_k,t_k)$ is the 6x6 identity matrix I_6. The solution to equation (6-19) is

$$\Phi(t_{k+1},t_k) = e^{\int_{t_k}^{t_{k+1}} F(t)dt}$$

$$= I_6 + \int_{t_k}^{t_{k+1}} F(t)dt + 1/2\left[\int_{t_k}^{t_{k+1}} F(t)dt\right]^2 + \ldots . \qquad (6\text{-}49)$$

Then, since $\int_{t_k}^{t_{k+1}} F(t)dt$

$$= \begin{bmatrix} 0 & 0 & 0 & (t_{k+1}-t_k) & 0 & 0 \\ 0 & 0 & 0 & 0 & (t_{k+1}-t_k) & 0 \\ 0 & 0 & 0 & 0 & 0 & (t_{k+1}-t_k) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\left(\int_{t_k}^{t_{k+1}} F(t)dt \right)^n$ is the zero matrix for n>1,

$$\Phi(t_{k+1},t_k) = \begin{bmatrix} 1 & 0 & 0 & (t_{k+1}-t_k) & 0 & 0 \\ 0 & 1 & 0 & 0 & (t_{k+1}-t_k) & 0 \\ 0 & 0 & 1 & 0 & 0 & (t_{k+1}-t_k) \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6\text{-}50)$$

(4) The next calculation is equation (6-20) to obtain the state noise covariance matrix Q*. To evaluate equation (6-20), compute the product $\Phi(t_{k+1},t)G(t)Q(t)G^T(t)\Phi^T(t_{k+1},t)$. Recall that in defining the dynamic model, it was assumed $Q(t) = \sigma^2 I_3$. Furthermore, the matrix G may be partitioned as $\left[\begin{array}{c} 0_3 \\ \hline I_3 \end{array}\right]$,

where 0_3 is the 3x3 zero matrix. Then

$$G(t)Q(t)G^T(t) = \left[\begin{array}{c} 0_3 \\ \hline I_3 \end{array}\right] \left[\sigma^2 I_3 \right] \left[0_3 \mid I_3 \right]$$

$$= \left[\begin{array}{c|c} 0_3 & 0_3 \\ \hline 0_3 & \sigma^2 I_3 \end{array}\right] . \qquad (6\text{-}51)$$

Then, if $D = t_{k+1} - t$, then

$$\Phi(t_{k+1},t)G(t)Q(t)G^T(t) = \left[\begin{array}{c|c} I_3 & DI_3 \\ \hline 0_3 & I_3 \end{array}\right] \left[\begin{array}{c|c} 0_3 & 0_3 \\ \hline 0_3 & \sigma^2 I_3 \end{array}\right]$$

$$= \left[\begin{array}{c|c} 0_3 & \sigma^2 DI_3 \\ \hline 0_3 & \sigma^2 I_3 \end{array}\right] ,$$

and

$$\Phi(t_{k+1},t)G(t)Q(t)G^T(t)\Phi^T(t_{k+1},t) = \left[\begin{array}{c|c} 0_3 & \sigma^2 DI_3 \\ \hline 0_3 & \sigma^2 I_3 \end{array}\right] \left[\begin{array}{c|c} I_3 & 0_3 \\ \hline DI_3 & I_3 \end{array}\right]$$

$$= \left[\begin{array}{c|c} \sigma^2 D^2 I_3 & \sigma^2 DI_3 \\ \hline \sigma^2 DI_3 & \sigma^2 I_3 \end{array}\right] .$$

(6-52)

To evaluate equation (6-20), the integrals of the four blocks of the partitioned matrix (6-52) are calculated. For every nonzero element in the block $\sigma^2 D^2 I_3$,

$$\int_{t_k}^{t_{k+1}} \sigma^2 D^2 dt = \int_{t_k}^{t_{k+1}} \sigma^2 (t_{k+1} - t)^2 dt$$

$$= \sigma^2 \int_{t_k}^{t_{k+1}} (t_{k+1}^2 - 2t_{k+1}t + t^2) dt$$

$$= \sigma^2 (t_{k+1}^2 t - t_{k+1} t^2 + t^3/3) \Big|_{t_k}^{t_{k+1}}$$

$$= \sigma^2 (t_{k+1}^3 - t_{k+1}^2 t_k - t_{k+1}^3 + t_{k+1} t_k^2 + t_{k+1}^3/3 - t_k^3/3)$$

$$= \sigma^2(t_{k+1}^3/3 - t_{k+1}^2 t_k + t_{k+1}t_k^2 - t_k^3/3)$$

$$= \frac{\sigma^2}{3}(t_{k+1} - t_k)^3. \qquad (6\text{-}53)$$

Hence,

$$\int_{t_k}^{t_{k+1}} \sigma^2 D^2 I_3 \, dt = \frac{\sigma^2}{3}(t_{k+1} - t_k)^3 I_3. \qquad (6\text{-}54)$$

Similarly,

$$\int_{t_k}^{t_{k+1}} \sigma^2 D I_3 \, dt = \frac{\sigma^2}{2}(t_{k+1} - t_k)^2 I_3,$$

and

$$\int_{t_k}^{t_{k+1}} \sigma^2 I_3 \, dt = \sigma^2(t_{k+1} - t_k) I_3. \qquad (6\text{-}55)$$

Thus,

$$Q^*(t_{k+1}) = \sigma^2 \begin{bmatrix} \frac{(1/3)(t_{k+1}-t_k)^3 I_3}{(1/2)(t_{k+1}-t_k)^2 I_3} & \Big| & \frac{(1/2)(t_{k+1}-t_k)^2 I_3}{(t_{k+1}-t_k) I_3} \end{bmatrix}. \qquad (6\text{-}56)$$

(5) This step defines the measurement matrix for the measurements at time t_{k+1}. In processing the measurements individually, for each t_{k+1} there are N matrices $H_i(t_{k+1})$, i=1,..., N. Each H_i is defined by equation (6-21) as

$$H_i(t_{k+1}) = \left. \frac{\partial h_i}{\partial \underline{x}} \right|_{\underline{x} = \hat{\underline{x}}\,(t_{k+1}/t_k)}, \qquad (6\text{-}57)$$

where h_i is defined in equation (6-32) and $\hat{\underline{x}}\,(t_{k+1}/t_k)$ was obtained in step (1). The matrix H_i is defined by equation (6-57) to be 1x6. The first component of H_i is

$$\frac{\partial h_i}{\partial x(t)} = \frac{\partial\left(\sqrt{(x(t) - X_i)^2 + (y(t) - Y_i)^2 + (z(t) - Z_i)^2}\right)}{\partial x(t)}$$

$$= \frac{x(t) - X_i}{\sqrt{(x(t) - X_i)^2 + (y(t) - Y_i)^2 + (z(t) - Z_i)^2}}$$

which, evaluated at $\hat{\underline{x}}(t_{k+1}/t_k)$ is

$$\frac{(\hat{x}(t_{k+1}/t_k) - X_i)}{\sqrt{(\hat{x}(t_{k+1}/t_k) - X_i)^2 + (\hat{y}(t_{k+1}/t_k) - Y_i)^2 + (\hat{z}(t_{k+1}/t_k) - Z_i)^2}}. \tag{6-58}$$

Similar calculations can be made for y and z. Since h_i is not a function of vx, vy, or vz, then

$$\frac{\partial h_i}{\partial vx} = \frac{\partial h_i}{\partial vy} = \frac{\partial h_i}{\partial vz} = 0 \quad \text{for } i = 1,\ldots,N. \tag{6-59}$$

Hence, if r denotes the denominator of equation (6-58), the measurement matrix H_i, i = 1,..., N, is

$$H_i(t_{k+1}) = \left[\frac{(\hat{x}(t_{k+1}/t_k) - X_i)}{r} \quad \frac{(\hat{y}(t_{k+1}/t_k) - Y_i)}{r} \quad \frac{(\hat{z}(t_{k+1}/t_k) - Z_i)}{r} \quad 0 \; 0 \; 0\right] \tag{6-60}$$

Now all the values have been obtained that are needed to update the estimate of the position and velocity of the target at time t_{k+1}, given the estimate at time t_k, and the measurements at time t_{k+1}. Assuming that the initial values $\hat{x}(t_0)$ and P(0) are given, then the following sequence will give the Kalman filter for this example. This sequence can be used to write the filter program, in FORTRAN for example, with steps (1) through (5) as subroutines which are called at the appropriate times in the sequence.

First, extrapolate the state vector $\hat{\underline{x}}(t_k/t_k)$ using equation (6-42)in step (1). Also extrapolate the state covariance matrix $P(t_k/t_k)$ using equation (6-22). For this, $I(t_{k+1},t_k)$ is needed from step (3) and $Q^*(t_{k+1})$ from step (4).

The data at time t_{k+1} is input as an N-vector $\underline{y}(t_{k+1})$. However, process the measurement from each station individually. To do so, process them in sequence by considering each measurement y_i, i=1,...,N as a one-dimensional measurement vector. Then go from y_i to y_{i+1} in a similar manner as going from $\underline{y}(t_k)$ to $\underline{y}(t_{k+1})$. Once the calculations of all the y_i's have been completed, the measurement vector for time t_{k+1} will be known, which is designated as

$$\underline{y}(t_{k+1}) = (y_1, y_2, \ldots. y_N). \qquad (6\text{-}61)$$

Now, begin the sequence of measurement updates for time t_{k+1} with y_1.

The gain matrix is computed using equation (6-23). The matrix H_1 from step (5) is used here. Note then that the resulting $K(t_{k+1})$ will be the first of N gain matrices, denoted $K_i(t_{k+1})$, i = 1,...,N and that they are all 6x1.

The measurement update of equation (6-24) is then completed for the one-dimensional vector y_1, using y_1 as the $y(t_{k+1})$ of equation (6-24), and plugging $h_1(\hat{\underline{x}}(t_{k+1}/t_k))$ from equation (6-32) into the $\underline{h}(\hat{\underline{x}}(t_{k+1}/t_k),t_k)$ of equation (6-24). The output $\hat{\underline{x}}(t_{k+1}/t_{k+1})$ will be designated as $\hat{\underline{x}}_1(t_{k+1}/t_{k+1})$ to indicate its calculation from measurement y_1. The covariance matrix is also updated, using equation (6-25) and resulting in $P_1(t_{k+1}/t_{k+1})$.

Only the last three steps consisting of gain calculation, measurement update, and covariance update are repeated for measurements $y_2, y_3, \ldots, y_N$. No time update is needed because these measurements are all at the same time as y_1. Thus, for measurement y_2, $\hat{\underline{x}}(t_{k+1}/t_k)$ is replaced in equation (6-24) by $\hat{\underline{x}}(t_{k+1}/t_{k+1})$, and $P(t_{k+1}/t_k)$ is replaced in equations (6-23) and (6-25) by $P_1(t_{k+1}/t_{k+1})$, since $\hat{\underline{x}}_1$ and P_1 are the best estimates currently available. The output of equation (6-24) will then be $\hat{\underline{x}}_2(t_{k+1}/t_{k+1})$ and that of equation (6-25) will be $P_2(t_{k+1}/t_{k+1})$. These three steps are repeated until $\hat{\underline{x}}_N(t_{k+1}/t_{k+1})$ is obtained which is the estimate of the target's position and velocity at time t_{k+1} given all the measurements up to and including those at time t_{k+1}, and $P_N(t_{k+1}/t_{k+1})$ which is its corresponding covariance matrix. The updates have now been completed for the $(k+1)^{st}$ time point.

CHAPTER 7

FILTERS IN CURRENT RANGE USE

This chapter contains a description of some of the most frequently used filters. The filters sampled from the survey provide a good cross section of range data applications, and no attempt has been made to select a filter which is best for any application. There are seven least squares filter applications in use identified by the survey, and probably several more that were not identified. Discussion of several variations of the least squares filters is given in this chapter. Also discussed is the Quadratic Digital (QD) filter which has been in use for many years, and at the time of the survey, there were five QD users. The originator of the QD filter is Mr. W. A. McCool, White Sands Missile Range (WSMR). A section on the Digital Filter X (DFX) is included to provide support for the QD filter.

Appendix A contains a summary of filter information based on a survey of the Data Reduction and Computer Group of the Range Commanders Council. The appendix essentially contains all of the information received. In addition, the appendix provides areas of application used at the contributing ranges and observations about the suitability of the filters.

7.1 Least Squares Filters

There are numerous filters based on least squares. In this section, one basic and most widely used least squares filters is the least squares polynomial moving arc filter. Although the simple average filter is not usually recognized as a type of least squares filter, it is addressed here. The least squares methods described throughout the rest of the chapter are variations of the least squares polynomial moving arc filter.

7.1.1 Simple Moving Average Filters

Simple moving average filters give the average of the N most recent observations and are equivalent to fitting a zeroth degree polynomial evaluated at the midpoint. The formulation is quite elementary; it is

$$M_t = \frac{x_t + x_{t-1} + \cdots + x_{t-N+1}}{N} \qquad (7\text{-}1)$$

$$= M_{t-1} + \frac{x_t - x_{t-N}}{N}.$$

This filter is very simple and straightforward and is useful for observing a constant process. However, if the process is changing, a small value of N is needed for rapid response.

The following is an explanation of a simple moving average filter as a least squares filter. It is assumed that the average is a constant value (call it a) expressed in terms of the parameter in consideration (call it x). The problem starts out with the simple equation

$$x = a \tag{7-2}$$

The problem now is to find the value for a so that the N data points $x_{t-N+1},\ldots,x_t$ collectively deviate from equation (7-2) as little as possible (in the least squares sense). Then, to find the best least squares estimate for a over the last N values of x, the value

$$S = \sum_{i=t-N+1}^{t} (x_i-a)^2 \tag{7-3}$$

must be minimized.

Taking the partial derivative of S with respect to a,

$$\frac{\partial S}{\partial a} = 2 \sum_{i=t-N+1}^{t} (x_i - a) = 0 \tag{7-4}$$

It can be seen that S is minimized when

$$\sum_{i=1}^{t} x_i = na \tag{7-5}$$

from which it is found that the best least squares estimate for a is

$$\frac{\sum_{i=t-N+1}^{t} x_i}{N} \tag{7-6}$$

the average value of the parameter at the most recent N points.

7.1.2 Least Squares Polynomial Moving Arc Filters

Least squares polynomial moving arc filters are based on the assumption that the true function can be expressed over a finite

span of time by a polynomial of fairly low degree and that the errors in the measurements are random with zero mean and finite rms value and are serially uncorrelated. Noted here, the smaller the time intervals are, the less the degree the polynomial needs to be to provide a good estimating curve for the data. A sampling interval of .05 seconds is said to be sufficiently small for good data fitting. It is desired to solve for the coefficients of the polynomial which best fits the data in a least squares sense, that is, to minimize the sum

$$S = \sum_{i=1}^{n} (a_0 + a_1 t_i + a_2 t_i^2 + \ldots + a_d t_i^d - y_i)^2 \qquad (7\text{-}7)$$

where

a_k, k = 0, 1,...,d are the coefficients of the polynomial of degree d

t_i = time of each sample referenced to the midpoint of the span under consideration.

Y_i = sampled measurements

n = number of points in the span (constrained to be odd and greater than d).

To minimize S, take the partial derivatives of S with respect to each of the unknowns, a_k, and solve the resulting set of d+1 linear equations. In matrix notation, write this as

$$CA = B, \text{ where} \qquad (7\text{-}8)$$

$$C \text{ is composed of } C_{ij} = \sum_{k=1}^{n} t_k^{i+j-2}, \quad \begin{matrix} i=1, 2,\ldots,d+1, \\ j=1, 2,\ldots,d+1 \end{matrix}$$

$$A = (a_0, a_1,\ldots,a_d)^T, \text{ and}$$

$$B \text{ is a vector with elements } b_i = \sum_{k=1}^{n} y_k t_k^{i-1}, \; i=1, 2,\ldots,d+1.$$

Then $A = C^{-1}B$. If the polynomial $a_0 + a_1 t + a_2 t^2 + \ldots + a_d t^d$ represents position, then velocity and acceleration are found by taking the derivatives of the polynomial with respect to time.

The vector A of coefficients has been found for the first n time points. Normally, the polynomial with these coefficients is applied just to the midpoint of the time span. The first (n-1)/2 points are generally used just for finding the new data value for

the midpoint, but new values may be generated (using that polynomial) for the first (n-1)/2 points also. Once the new midpoint data value has been found, the next step is to redetermine the coefficients of the polynomial using points 2 to n+1, so a new data value for the new midpoint can be found. Repeat this process until finished with the polynomial moving from points m,...,m+n-1 to m+1,...,m+n with each step (where m ranges from 1 to the total number of time points minus n), thus describing a moving arc.

If the data are sampled at equally spaced intervals of time, the t can be considered to be unity and C^{-1} is uniquely determined when d and n are chosen. By use of C^{-1} and t_i, a set of weights can be computed for determining the a_k such that

$$a_k = \sum_{i=1}^{n} w_{k\,i}\, y_{N-n+1,} \text{ where} \qquad (7\text{-}9)$$

N is the subscript of the last point in the span.

These weights are useful for determining the expected error in position, velocity, and acceleration for data with known input error since

$$\frac{\sigma_{a_k}}{\sigma_y} = \left(\sum_{i=1}^{n} w_{ki}2 \right)^{1/2}. \qquad (7\text{-}10)$$

It is assumed that the standard deviations of all the measurements y_i are the same, and it is called σ_y. Thus the sampling interval can be chosen, as well as the number of points smoothing and the degree of the polynomial which will accomplish the desired results, with a minimum expense of computational effort (see reference 3). There are methods of calculating C^{-1} that do not entail finding the inverse directly, which would be very time consuming. One of these methods will be discussed later in this chapter.

An advantage of using a polynomial is that much need not be known about the true process, because if the polynomial is of high enough degree, it will tend to seek out the signal in the presence of noise and give a reasonable estimate of it. For this reason, the higher degree of the polynomial, the more flexibility is allowed in accurately constructing the true process. Another reason is by using least squares, the polynomial is not forced to pass exactly through any of the observations, which would result in a certain amount of smoothing.

7.2 Position and Velocity Constrained Least Squares Filters

Position and velocity constrained least squares filters are based on the philosophy that the best estimate of the true function and its derivatives can be obtained by fitting a polynomial to fixed time spans of the measurements by the method of least squares while imposing constraints on the polynomial which force continuity between successive spans. The assumptions concerning the data are the same as for least squares polynomial filters, but the additional requirement is made that the polynomial being fit to the filter span must pass through a point on the previous curve with the same slope. For example, consider fitting a quadratic to data sampled at evenly spaced intervals. Suppose there are five points in the span, that the midpoint of the span is at t=0, and that each point in the span is one unit apart from the one next to it. For the first span, the time values are t_i, i=1,...,5 where t_i= i-3. It would be desirable to have

$$\hat{y}_i = a_0 + a_1 t_i + a_2 t_i^2 \qquad (7\text{-}11)$$

evaluated at i=3 to be the filtered data value for the midpoint. Thus, the filtered data value is equated with a_0, since $t_3 = 0$. Once a_0, a_1, and a_2 for this span have been solved for using the regular least squares method, the first step is completed.

The next step is to solve for $\hat{y}_i$ at $t_i = 1$. Move the span one point over so that the new span is for points i=2,...,6. The new midpoint is at i=4. Remember there are two constraints: position and velocity. Note now that t_4 is the middle time value for the new span. The idea of the filter is that the polynomials of both spans have the same value (position) and the same slope (velocity) at time t_4.

For both the old and new spans, consider the middle time value to be 0 and the following time value to be 1. So for the old time span, $t_3 = 0$ and $t_4 = 1$. For the new time span, $t_3 = -1$, $t_4 = 0$, and $t_5 = 1$. Let

$$p(t_4) = a_0' + a_1' t_4 + a_2' t_4^2 \qquad (7\text{-}12)$$

be the polynomial for the midpoint of the new span, and let

$$q(t_4) = a_0 + a_1 t_4 + a_2 t_4^2 \qquad (7\text{-}13)$$

be the polynomial for the same point of the old span. Use the position constraint to obtain a_0' and the velocity constraint to obtain a_1'.

7.2.1 Position Constraint

Because $p(t_4)$ is for the midpoint of the new span, the method equates t_4 to 0, so that $p(t_4)$ takes on the value a_0' Because $q(t_4)$ is for the point to the right of the midpoint of the old span, the method equates t_4 to 1, so that $q(t_4)$ assumes the value $a_0 + a_1 + a_2$. To ensure that these two polynomials have the same value at t_4, equate a_0' with $a_0 + a_1 + a_2$. Now the value has been found for the first coefficient for the new span.

7.2.2 Velocity Constraint

Again, consider the polynomial $p(t_4)$ for the midpoint of the new span. The slope for this polynomial is $a_1' + 2a_2't_4$. Because this slope is for the midpoint of the new span, the method evaluates this slope at $t_4 = 0$, whence the slope takes on the value a_1'. Also consider the polynomial $q(t_4)$ for the same point of the old span. The slope for this polynomial is $a_1 + 2a_2t_4$. Because this slope is for the midpoint of the old span, the method evaluates the slope at t=1, whence the slope assumes the value $a_1 + 2a_2$. To ensure that the slopes of these two polynomials have the same value at t_4, we equate a_1 with $a_1 + 2a_2$. Now the value has been found for the second coefficient of the polynomial for the new span.

So from the position constraint,

$$a_0' = a_0 + a_1 + a_2 \tag{7-14}$$

and from the velocity constraint,

$$a_1' = a_1 + 2a_2. \tag{7-15}$$

Now proceed to find the value for a_2'.

Start by minimizing the sum

$$S = \sum_{i=N-n+1}^{N} (a_0'+a_1't+a_2't_i^2 - y_i)^2 \tag{7-16}$$

where

a_0', a_1', and a_2' are coefficients for the second span

n = number of points in the span (should be odd and greater than 2; in this case, n=5)

N = subscript of last point in the span (in this case, N=6)

t_i = time of each sample referenced to the midpoint of the span such that $t_i - t_{i-1} = 1$

y_i = sampled measurements.

Taking the partial derivative of S with respect to the unknown, a_2, equating it to zero, and solving for a_2' gives

$$a_2' = \frac{1}{\Sigma t_i^4} \left(\Sigma\, y_i t_i^2 - a_0' \Sigma\, t_i^2 - a_1' \Sigma\, t_i^3\right). \qquad (7\text{-}17)$$

Similarly to the first span, the filtered data value for the midpoint of the second span is

$$\hat{y}_i = a_0' + a_1' t_i + a_2' t_i^2 \qquad (7\text{-}18)$$

evaluated at i = 4. That is, equate the filtered data value with a_0, since $t_4 = 0$. The subsequent steps are done in the same manner as the second step. Sometimes it is desirable to constrain position only to be obtained from the previous fit. In this case both a_1' and a_2' in equation (7-16) are unknown, and the solution would involve two equations with two unknowns.

This method has a distinct computational advantage over unconstrained least squares, because there are fewer summations and simpler equations. It also produces smoother output because of its recursive nature. However, it is slower to react to a change in the input and under certain conditions results in oscillation in the output.

7.3 Orthogonal Polynomials

In smoothing data by the usual method of least squares, it is necessary to choose in advance the degree of polynomial which will be used to approximate the data. This choice is necessary because the coefficients found are dependent upon the degree of curve being fitted. Often, however, it is not known in advance what degree curve will best fit the data. In such a case, it is desirable to fit several polynomials, each time increasing the degree used, until it is seen that any further increases would not produce a significantly better fit. The computation of successive polynomials is greatly simplified by the use of the orthogonal polynomial procedure. This method determines the approximating polynomial in terms of another variable, so chosen that each coefficient found is independent of the others, making it possible to increase the degree of curve used without making it necessary to recompute the previously found coefficients.

This program is generally used to smooth position data. The degree of curve fitted is increased until an F-test indicates that additional coefficients of the polynomial would not be significantly different from zero. The smoothed positions are then differentiated to obtain velocities, and the velocities differentiated to obtain accelerations. The error estimates of the smoothed data and derivatives are computed in the form of standard deviations for each point. Coefficients of the original polynomial are derived in terms of the new polynomial. Orthogonal polynomials are discussed in reference 46. The method is given here. First, the term "orthogonal polynomials" is defined.

Two polynomials $P_i(x)$ and $P_j(x)$ (of degrees i and j) are orthogonal on a set of points $x_1,\ldots,x_n$ provided that

$$\sum_{k=1}^{n} P_i(x_k) P_j(x_k) = 0 \qquad \text{for } i \neq j. \tag{7-19}$$

Given a polynomial of order k, k orthogonal polynomials can be found whose linear combination expresses the original polynomial. The method of orthogonal polynomials follows. Assume that the polynomial

$$p(t_i) = a_0 + a_1 t_i + a_2 t_i^2 + \ldots + a_k t_i^k \tag{7-20}$$

expresses the true function. Reference 46 shows directly and by example that this polynomial may be re-expressed as

$$p(t_i) = b_0 P_{0,t_i} + b_1 P_{1,t_i} + \ldots + b_2 P_{2,t_i}$$

where $P_{0,t_i} = 1$

$$P_{1,t_i} = t_i$$

$$P_{j,t_i} = t_i P_{j-1,i} - \frac{Q_{j-1}}{Q_{j-2}} P_{j-2,i}$$

$$b_j = \frac{y_i P_{j,t_i}}{Q_j}$$

$$\text{where } Q_j = \sum_i P_{j\,t_i}^2 \tag{7-21}$$

The time variable t_i ranges from $-(N-1)/2$ to $(N-1)/2$ in steps of one. (The index i could have been used instead of t_i, but t is the more conventional variable for time.) Reference 46 shows that the P_{j,t_i}'s are orthogonal to each other.

Once the original polynomial of equation (7-20) has been re-expressed in terms of orthogonal polynomials (in equation (7-21)), p(t) can be expanded to be of one more degree through the Gram-Schmidt orthogonalization process,

where

$$\sum_{k=1}^{n} P_i(x_k)P_j(x_k) \text{ is the inner product and} \qquad (7\text{-}22)$$

where the polynomials $P_i(x)$ and $P_j(x)$ are vectors in the vector space of polynomials spanned by $1,x,x^2,\ldots,x^n$, where n is the degree of the higher degree polynomial. In this manner, the values of any of the previously obtained b_j's or $P_{j,t}$'s are not changed. Instead, another such term is merely added on.

At what degree should expanding stop? Reference 46 suggests a clever criterion for deciding when to stop. The variance V of the data can be expressed in the following two ways:

$$V = \sigma_a^2 = \frac{\sum_{i=-(N-1)/2}^{(N-1)/2} \left(\sum_{j=1}^{k} a_j t_i^j - y_i\right)^2}{N - 1 - k}$$

$$V = \sigma_b^2 = Q_j(b_j - B_j)^2 \qquad (7\text{-}23)$$

where b_j is the computed coefficient and B_j is the true coefficient. Reference 46 gives the derivations for the last two equations. Of course, $\sigma_a^2 = \sigma_b^2$, but two different notations are given for the sake of future reference. The object now is to test the hypothesis that the true value of the jth coefficient B_j is zero. Test the hypothesis using an F-test. (The F-test tests the proportion of two variances.) Test the proportion,

$$\frac{\sigma_b^2\Big|_{B_j 0}}{\sigma_a^2} \qquad (7\text{-}24)$$

Of course, the numerator $\sigma_b^2\Big|_{B_j=0} = Q_j b_j^2$. If B_j is ideally 0, then this proportion should be close to 1.

In the test, if

$$\frac{\sigma_b^2\Big|_{B_j=0}}{\sigma_a^2} \leq F, \qquad (7\text{-}25)$$

then the hypothesis is accepted as true, and b_j is set equal to zero. The next coefficient is tested in the same manner. When two consecutive coefficients are set equal to zero, then the degree k of the polynomial is determined from the last coefficient which was not set equal to zero; that is, k=j-2.

Stopping after two consecutive zero coefficients assumes that succeeding terms in the polynomial will carry negligible weight. For practical purposes, most parameters can be modeled using a lower-order polynomial. If the polynomial is of higher degree, the higher coefficients are close to zero. Normally, data generally does not shoot up or down suddenly enough to warrant higher degree, high- magnitude coefficients.

7.4 Least Squares Polynomial Moving Arc Filters Using Recursive Sums

A recursive scheme for computing the sums required to perform least squares polynomial moving arc smoothing has been developed at the Air Force Development Test Center which permits this type of smoothing to be accomplished considerably faster than by the usual linear combinations of observations. This is not a recursive filter in the usual sense in that previous filter output is not used in determining current output.

In paragraph 7.1 it was stated that the equation for this type of smoothing may be written in matrix notation as

$$A = C^{-1}B, \qquad (7\text{-}26)$$

where the elements of A, B, and C have been previously defined. Each element of B and C consists of a summation of n terms (n = number of points smoothing) which must be performed each time a point is dropped and another is picked up in the moving arc

process. However, these summations can be avoided by computing the sums for a given span in terms of those for the previous span.

For instance, suppose it was found that through the method discussed in paragraph 7.1

$$b_i^{(1)} = \sum_{k=1}^{n} y_k \, t_k^{i-1}, \qquad t_k = T_k - T_{\frac{n+1}{2}}, \tag{7-27}$$

where T_i is the i^{th} time value. So t_k is the length of the interval between the k^{th} time point and the midpoint of the span. To find

$$b_i^{(2)} = \sum_{k=2}^{n+1} y_k \, (t_k - \Delta)^{i-1}, \text{ where } \Delta = T_{\frac{n+3}{2}} - T_{\frac{n+1}{2}}, \tag{7-28}$$

do so in terms of $b_i^{(1)}$ by means of the recursion

$$b_i^{(2)} = \sum_{j=0}^{i-1} (-\Delta)^j \binom{i-1}{j} b_{i-j}^{(1)} + y_{n+1} \sum_{j=0}^{i-1} (-\Delta)^j \binom{i-1}{j} t_{n+1}^{i-j-1} \tag{7-29}$$

$$- y_1 \sum_{j=0}^{i-1} (-\Delta)^j \binom{i-1}{j} t_1^{i-j-1} .$$

The elements of C are computed similarly by considering the y_k=1. However, as previously stated, if the data are evenly spaced, C is uniquely determined when the degree of the polynomial and number of points smoothing have been selected. In this case, C^{-1} can be precomputed and stored for use in the smoothing. In addition, Δ can be considered unity, and the equations for the recursion are simplified.

With this formulation no lengthy summations are required and the number of operations, once initiated, is independent of the number of points used in the smoothing. Furthermore, by computing recursively $\sum . y_k^2$, reference 63 shows that with very little extra effort the residual sum of squares of each span can be obtained as an estimate of the error in the data by

$$s = y_k^2 - A^T B. \tag{7-30}$$

7.5 Derivative Information Recovery by a Selective Integration Technique (DIRSIT)

The data smoothing technique known as Derivative Information Recovery by a Selective Integration Technique (DIRSIT) was developed in 1962 at White Sands Missile Range. The technique has been modified somewhat to permit more control of interval program parameters to satisfy the requirements of a greater number of users. The basic steps in the DIRSIT smoothing philosophy are

(1) the sign of the difference between the raw and smoothed position data is examined at each point within the filter span;

(2) the smoothed position data is satisfactory if the number of positive signs differs from the number of negative signs by one; (The total number of signs is odd.) and,

(3) if (2) above is not satisfied, the acceleration history within the filter span is modified as much as necessary to force the difference to one.

It is assumed that sampled position data is being entered with associated time. The output is smoothed position, velocity, and acceleration with associated time. For the filter span 2m+1, which is odd, table 7-1 shows a starting condition.

TABLE 7-1. A STARTING CONDITION FOR A DIRSIT FILTER WHOSE SPAN IS 2m+1 POINTS

Raw Position	Smoothed Position	Velocity	Acceleration
X_1	$\bar{X}_1$	$\dot{X}_1$	$\ddot{X}_1$
X_2			$\ddot{X}_2$
.			.
.			.
.			.
$Xm+1$			$\ddot{X}m+1$
.			.
.			.
.			.
X_{2m+1}			$\ddot{X}_{2m+1}$

The parameters $\bar{x}_1$, $\dot{x}_1$, x_i, i=1,...,2m+1 are generated during initialization; consequently, the initial estimates are obtained independently. Columns $\bar{x}_i$ and x_i are computed as

$$\bar{x}_{i+1} = \bar{x}_i + \Delta t_{i+1}\dot{x}_i + (1/2)\,\Delta\, t_i^2\ddot{x}_1 \qquad (7\text{-}31)$$

$$\dot{x}_{i+1} = \dot{x}_i + \Delta t_{i+1}\ddot{x}_1 \qquad (7\text{-}32)$$

where i=1,...,2m+1; $\Delta t_{i+1} = t_{i+1} - t_i$

All accelerations $\ddot{x}_i$,i=m+1,...,2m+1 are equal.

Having completed the columns of table 7-1, examine how closely the smoothed and raw positions agree. Generate m values of xi by computing $\Delta x_i = x_i - x_i$, i=m+1,...,2m+1. Hence, the number of negative Δx_i and the number of nonnegative Δx_i cannot both be odd, neither can they both be even. Let NL be the number of Δx_i which are negative; that is, the $\bar{x}_i$ is larger than the x_i. Let NS be the number of Δx_i which are nonnegative. NL + NS = m and the Δx_i (i=m+2,...,2m+1) are acceptable if $|\,NL-NS\,| \geq 3$. These conditions are corrected by changes in the $\ddot{x}_i$, i=m+1,...,2m+1, which will also produce changes in the $\dot{x}_i$ and $\bar{x}_i$, i=m+2,...,2m+1.

Examine the procedure for changing the $\ddot{x}_i$. To be specific, assume that NS $\geq$ NL + 3; then the x_i must be increased. Starting at some time t, the acceleration will be increased with slope 1, to time t_{m+1} which will be called t_p, the pivot point. Then for times t_p, $t_{m+2},\ldots,t_{2m+1}$ the acceleration increase will be constant. This type of increase in acceleration will just result in a change in only one $\bar{x}_k$ so that Δx_k would just barely become negative where formerly it was nonnegative. The first step is to equate Δx_k as shown in the following equation:

$$\Delta x_k = (1/6)(t_p-\bar{t})^3 + (1/2)(t_p-\bar{t})^2(t_k-t_p) + (1/2)(t_p-\bar{t})(t_k-t_p)^2 \qquad (7\text{-}33)$$

Note that the change in distance is obtained by a double integration of acceleration. First integration of figure 7-1 would yield a velocity curve of the shape shown in figure 7-2.

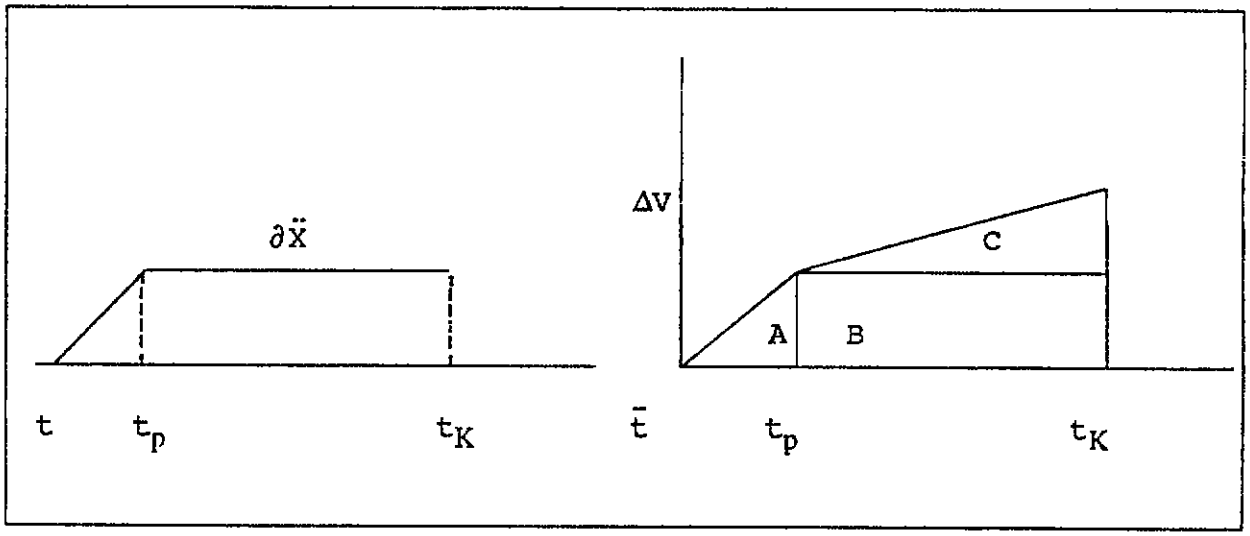

Figure 7-1. Example acceleration curve.

Figure 7-2. Corresponding velocity curve.

Where Area A = $(1/6)(t_p-\bar{t})^3$, Area B = $(1/2)(t_p-\bar{t})^2(t_k-t_p)$,

and Area C = $(1/2)(t_p-\bar{t})(t_k-t_p)^2$ (7-34)

$$\Delta x_k = (1/6)\left[(t_p-\bar{t})^3 + 3(t_p-\bar{t})^2(t_k-t_p) + 3(t_p-\bar{t})(t_k-t_p)^2\right] \quad (7\text{-}35)$$

$$= (1/6)\left[(t_k-\bar{t})^3 - (t_k-t_p)^3\right]$$

from which $(t_p-\bar{t}) = [6\Delta x_k \div (t_k-t_p)^3]^{1/3} - (t_k-t_p)$ (7-36)

Thus, equation (7-36) gives the distance between t- and t_p. This change made in x_i and the changed x_k would decrease NS by 1 and increase NL by 1. For each positive Δx_j, compute a t_i but only the largest $\bar{t}_1$ is used. Thus only one Δx_i would change sign. For this selected value of $\bar{t}$, the changed $\ddot{x}_i$, $\dot{x}_i$, and x_i are calculated for each time beyond $\bar{t}$. A check is made again: if NL-NS$\leq$1, then the output at t, of $\bar{x}$, $\dot{x}$, $\ddot{x}$, is the output of DIRSIT; if NL-NS>1, the above computation of $\bar{t}$ is repeated.

Suppose that NL $\geq$ NS+3 or that the number of negative Δx_j is three or more greater than the number of nonnegative Δx_j. Under these conditions equation (7-36) becomes

$$(t_p-\bar{t}) = [-6\Delta x_k \div (t_k-t_p)]^{1/3} - (t_k-t_p) \quad (7\text{-}37)$$

To this point it has been assumed that $\bar{t} \leq t_1$, when $\bar{t} > t_1$ another procedure is employed. Increase the slope of the acceleration as shown in figure 7-3.

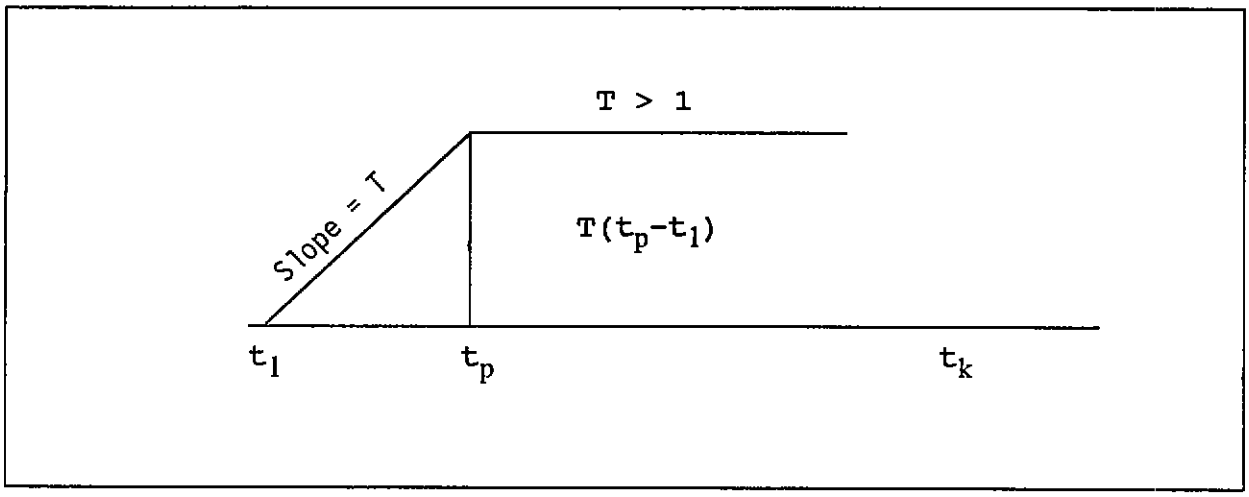

Figure 7-3. Example acceleration curve for $t > t_1$

$$T = \frac{6\Delta^x k}{(t_k - t_1)^3 - (t_k - t_p)^3} \qquad (7\text{-}38)$$

With this new value of T the new $\ddot{\bar{x}}_i$, $\dot{\bar{x}}_i$, $\bar{x}_i$, i=2,...,2m+1 are calculated. If | NL-NS | =1, the data output is finished; if not, repeat calculation of t. Now redefine terms by shifting their previous values down one, that is, shift

$$\begin{aligned} t_{i+1} &\longrightarrow t_i \\ x_{i+1} &\longrightarrow x_i \\ \bar{x}_{i+1} &\longrightarrow \bar{x}_i \\ \dot{\bar{x}}_{i+1} &\longrightarrow \dot{\bar{x}}_i \\ \ddot{\bar{x}}_{i+1} &\longrightarrow \ddot{\bar{x}}_i \end{aligned} \qquad i=2,\ldots,m \qquad (7\text{-}39)$$

The next data point is now input to the filter and the above process is repeated.

The DIRSIT was the original real-time flight filter at WSMR and has been successfully used with data from such missile systems as the Nike Zeus, ARPAT, Pershing, and Athena. The filter has been used for Nike Zeus and ARPAT elevation data, to

generate acquisition data, and to initialize a re-entry simulation in the Athena's real-time operational program. It is the major filter being used at Holloman Air Force Base and WSMR to evaulate the Athena re-entry data and is the prime source of post-flight data for the flight safety impact prediction programs at WSMR. The post-flight data is then used for vectoring recovery aircraft.

Because DIRSIT, a self-editing filter, uses a median-smoothing criterion, random spikes will have no effect on the filter output. Operation is not dependent upon equal time increments between data points, and a missing or repeated data point will not cause problems. With existing subroutines, numerous internal parameters can be controlled to obtain the desired filter responses. However, DIRSIT was not without its problems. Position, velocity, acceleration, and time arrays must be stored, thus requiring a significant amount of storage space and computing time. In addition, the filter has a lag which makes it undesirable for real-time use. (For additional information on DIRST, see references 36 through 39).

7.6 Digital Filter X (DFX)

In 1963 at WSMR, real-time flight safety support was first provided from the central computer facility. Originally, the DIRSIT filter was used, but it became evident that DIRSIT was not a satisfactory real-time filter because of the excessive storage requirements, computing time, and inherent lag. Consequently, the original version of the DFX filter was developed. The DFX is a variation of a constrained second degree filter which defines "best fit" criterion to be that the algebraic sum of the differences between the raw and smoothed data is zero.

The filter is more efficient in both speed and storage requirements because of a development which avoids computing the constraints explicitly and which stores only the array of differences rather than arrays of both raw and smoothed data. Additionally, a mean rather than a median smoothing criterion is used. The filter response is controlled by a set of input parameters which, in turn, determine the shape of the acceleration correction curve. The smoothing criterion for the DFX filter is

$$S = \sum_{i=1}^{N} (\bar{X}_i - X_i) = 0 \qquad (7\text{-}40)$$

which leads to

$$S = \sum_{i=1}^{N} (\bar{X}_0 + \dot{X}_0 (i\,\Delta t) + \ddot{X}_0 \frac{(i^2\,\Delta t^2)}{2} - X_i) = 0 \qquad (7\text{-}41)$$

With the constraints $\bar{X}_0$ and $\dot{X}_0$ fixed, only the acceleration has to be adjusted with each step

$$\ddot{X}_0 = \frac{1}{\frac{\Delta t^2 \sum i^2}{2}} \sum_{i=1}^{N} X_i - N\bar{X}_0 - \dot{X}_0 \Delta t \sum_{i=1}^{N} i \qquad (7\text{-}42)$$

Equation (7-41) differs from the constrained least squares (CLS) equations only in the multiplying factor $\frac{(i^2 \Delta t^2)}{2}$, and this difference will affect only the computation of the correction coefficients used by the filter. The method of applying the correction to the acceleration terms is the same for both a CLS and DFX filter.

To see how this method is developed, first look at the expression for $\hat{X}_{i+1}$ in terms of $\hat{X}_1$ where $\hat{x}_k$ is the new data value of x_k after smoothing:

$$\hat{X}_{i+1} = \hat{X}_1 + \hat{\dot{X}}_1 (i \Delta t) + \hat{\ddot{X}}_1 \frac{(i^2 \Delta t^2)}{2} \qquad (7\text{-}43)$$

(If x_{i+1} were expressed in terms of x_i, Δt would be the step size. In this case, consider i Δt to be the step size.) Redefine terms by shifting down the values for the next cycle, so that $\hat{x}_{j+1}$ of the previous step becomes what will be designated as $\hat{x}_j$ in the current step, where the bar indicates "prior to applying the correction factor."

This redefining of terms gives the following equation:

$$\bar{X}_i = \hat{X}_0 + \hat{\dot{X}}_0 (i \Delta t) + \hat{\ddot{X}}_0 \frac{(i^2 \Delta t^2)}{2} \qquad (7\text{-}44)$$

Compare the previous equation (7-44) with the following equation (7-45), which expresses the ith smoothed point based on the constraints.

$$\hat{X}_i = \hat{X}_0 + \hat{\dot{X}}_0 (i \Delta t) + \hat{\ddot{X}} \frac{(i^2 \Delta t^2)}{2} \qquad (7\text{-}45)$$

Subtracting equation (7-44) from equation (7-45)

$$(\hat{X}_i - \bar{X}_i) = \frac{(i^2 \Delta t^2)}{2} (\hat{\ddot{X}}_0 - \hat{\ddot{X}}_0) \qquad (7\text{-}46)$$

which will be written as

$$(\hat{X}_i - \bar{X}_i) = \frac{(i^2\ \Delta t^2)}{2}\ (\delta\ddot{X}_0) \qquad (7\text{-}47)$$

or

$$\hat{X}_i = \bar{X}_i + \frac{(i^2\ \Delta t^2)}{2}\ \delta\ddot{X}_0 \qquad (7\text{-}48)$$

That is, the smoothed position at the ith point equals the position before correction plus a constant factor depending only on the delta time increment and the position of the ith point in the N point span times the correction to the acceleration. The significance of this fact is that if there are predicted outputs and a means of computing the acceleration correction, then there is no need to compute the constraints explicitly or save arrays of both raw and smooth positions. Note that equation (7-48) can also be written

$$(\hat{X}_i - X_i) = (\bar{X}_i - X_i) + \frac{(i^2\ \Delta t^2)}{2}\ \delta\ddot{X}_0 \qquad (7\text{-}49)$$

or

$$\Delta X_i = \Delta\bar{X}_i + \frac{(i^2\ \Delta t^2)}{2}\ \delta\ddot{X}_0 \qquad (7\text{-}50)$$

The quantity $\sum_{i=1}^{N} \Delta X_i$ is set to be zero as the smoothing criterion.

The array ΔX_i can be used instead of the two arrays $\hat{X}_i$ and X_i for computing each cycle. The ΔX_i's, which are the differences between predicted and raw input at each time are saved, shifted down, and corrected after each succeeding computing cycle. As each ΔX_i is operated on in successive computing cycles, it becomes numerically smaller. Most of the correction is applied to the ΔX_i at the cycle when i=N, the real time span. This factor, it will be seen, is what makes the QD filter possible. Performance is almost identical to second order CLS smoothing; however DFX requires much less computation time and storage regardless of the span size. In addition, filter characteristics can be changed dynamically during flight to improve performance.

The basic DFX package includes editing, initialization, radar selection, and noise estimate routines. Versions of this filter were used in all WSMR real-time programs for over five years. References 40 and 41 are concerned with the DFX filter.

7.7 Quadratic Digital (QD) Filter

The DFX equations can be derived using the CLS smoothing criterion. If an approximation is made concerning the effects of the terms of the sums of products, the necessity of even one stored array is eliminated and the second order DFX formulation can be reduced to a set of six equations. The performance of the resulting (QD) filter is practically identical to CLS performance, although computing time is independent of the filter span. In reality, the QD filter span is implied rather than actual because no arrays are stored. The QD filter is essentially a DFX filter which applies all the corrections needed to the real-time point only.

The QD formulas employ a predict-correct procedure using the estimates determined for the current given data value evaluated in the preceding step. They are, therefore, said to be recursive. The prediction formulas are

$$\bar{\ddot{X}}_{n+1} = \hat{\ddot{X}}_n \tag{7-51}$$

$$\bar{\dot{X}}_{n+1} = \hat{\dot{X}}_n + \Delta t\, \hat{\ddot{X}}_n, \tag{7-52}$$

$$\bar{X}_{n+1} = \hat{X}_n + t\, \hat{\dot{X}}_n + \frac{\Delta t^2}{2}\, \hat{\ddot{X}}_n, \tag{7-53}$$

or

$$\bar{X}_{n+1} = \hat{X}_n + \frac{\Delta t}{2}\, (\bar{\dot{X}}_{n+1} + \hat{\dot{X}}_n) \tag{7-54}$$

where

$\bar{\ddot{X}}_{n+1}$ - is predicted second derivative,

$\hat{\ddot{X}}_n$ - second derivative estimate from preceding step,

$\bar{\dot{X}}_{n+1}$ - predicted first derivative,

$\hat{\dot{X}}_n$ - first derivative estimate from preceding step,

$\bar{X}_{n+1}$ - predicted data value,

$\hat{X}_n$ - data value estimate from preceding step, and

Δt - time differential from preceding step.

The corrections of the predicted values are based upon the differences between the given data and the predicted data values. The correction formulas are

$$\hat{\ddot{X}}_{n+1} = \bar{\ddot{X}}_{n+1} + K_1 \Delta X, \tag{7-55}$$

$$\hat{\dot{X}}_{n+1} = \bar{\dot{X}}_{n+1} + K_2 \Delta X, \tag{7-56}$$

$$\hat{X}_{n+1} = \bar{X}_{n+1} + K_3 \Delta X, \tag{7-57}$$

where

$\hat{\ddot{X}}_{n+1}$ - correction of the predicted second derivative,

$\hat{\dot{X}}_{n+1}$ - correction of the predicted first derivative,

$\hat{X}_{n+1}$ - correction of the predicted data value, and

K_1, K_2, K_3 - correction coefficients which minimize the error of the estimates:

ΔX - difference between given and predicted data values.

The functional relationship among the correction coefficients K_1, K_2, and K_3 are revealed through study of predicted values obtained by use of the second order CLS filter. In the CLS filter, intercept and slope constraints are applied at the oldest value end of a given data span M, so that the polynominal fit to the M data values in the least squares sense must also contain the estimates of the true value and the corresponding first derivative at the oldest span point. Using these constraints in the second order CLS filter, the predicted data value and its derivative can be obtained with the truncated Taylor series:

$$\bar{\dot{X}}_{n+1} = \hat{\dot{X}}_{n-M+1} + M\Delta t \; \hat{\ddot{X}}_{n-M+1}, \tag{7-58}$$

$$\bar{X}_{n+1} = \hat{X}_{n-M+1} + M\Delta t \; \hat{\dot{X}}_{n-M+1} + \frac{(M \Delta t)^2}{2} \; \hat{\ddot{X}}_{n-M+1}. \tag{7-59}$$

where n-M+1 indicates the oldest value in the filter span, having been shifted from its position in the preceding step of n-M+2. Since $\hat{X}_{n-M+1}$ and $\hat{\dot{X}}_{n-M+1}$ are fixed (because the position and velocity are constrained), only $\hat{\ddot{X}}_{n-M+1}$ can be adjusted to make equation (7-59) fit the M data values in the least squares sense. Let $\Delta\ddot{X}$ be the required correction. Then

$$\hat{\ddot{X}}_{n+1} = \hat{\ddot{X}}_{n-M+1} + \Delta\ddot{X} \tag{7-60}$$

Note that the second derivative is constant across the second order CLS filter span. Applying the correction to equations (7-58) and (7-59) holding x_{n-M+1} and $\dot{x}_{n-M+1}$ fixed, gives the estimates

$$\hat{\dot{X}}_{n+1} = \hat{\dot{X}}_{n-M+1} + M\,\Delta t(\hat{\ddot{X}}_{n-M+1} + \Delta\ddot{X}) \tag{7-61}$$

$$\hat{X}_{n+1} = \hat{X}_{n-M+1} + M\,\Delta t\hat{\dot{X}}_{n-M+1} + \frac{(M\,\Delta t)^2}{2}\,(\ddot{X}_{n-M+1} + \Delta\ddot{X}) \tag{7-62}$$

Subtracting equations (7-58) and (7-59) from equations (7-61) and (7-62) gives

$$\hat{\dot{X}}_{n+1} = \bar{\dot{X}}_{n+1} + M\,\Delta t\,\Delta\ddot{X}, \tag{7-63}$$

$$\hat{X}_{n+1} = \bar{X}_{n+1} + \frac{(M\,\Delta t)^2}{2}\,\Delta\ddot{X}. \tag{7-64}$$

Comparison of equations (7-60), (7-63), and (7-64) with equations (7-55), (7-56), and (7-57) reveals the relationship between K_1, K_2, and K_3 as desired. From equations (7-55) and (7-60)

$$\Delta\ddot{X} = K_1\Delta X \tag{7-65}$$

Then from equations (7-63), (7-56), and (7-65),

$$K_2 = M\Delta tK_1 \tag{7-66}$$

and from equations (7-64), (7-57), and (7-65),

$$K_3 = \frac{(M\Delta t)^2}{2}\,K_1 \tag{7-67}$$

In the QD theory, K_1 is developed as a function of the corresponding CLS filter span M, that is,

$$K_1 = K_1(M) = \frac{60M^2}{10M^3 + 33M^2 + 23M-6} \tag{7-68}$$

(See reference 45 for the derivation of this formula.) Thus, when M is arbitrarily specified, the QD correction coefficients are determined by equations (7-66), (7-67), and (7-68).

In its basic formulation, QD can be used as a real-time filter, because the argument (or subscript) of the estimates corresponds to the latest value accepted by the filter. On the other hand, estimates can be obtained for the data value at the oldest end of the span which are significantly better than the corresponding real-time estimates. Such estimates, which are often called "smoothed" data, correspond to the constraints computed in the CLS filter. Smoothed estimates are simply obtained with a Taylor series expanded about the real-time estimates, that is,

$$\hat{\hat{\hat{X}}}_{n+1} = \hat{\hat{X}}_{n+1} - (M-1)\ t\hat{\ddot{X}}_{n+1} \tag{7-69}$$

$$\hat{\hat{X}}_{n+1} = \hat{X}_{n+1} - \frac{(M-1)}{2}\ t\ (\hat{\hat{\hat{X}}}_{n+1} + \hat{\hat{X}}_{n+1}) \tag{7-70}$$

for the second order QD filter and

$$\hat{\hat{\ddot{X}}}_{n+1} = \hat{\ddot{X}}_{n+1} - (M-1)\ \Delta t\hat{\ddot{X}}_{n+1} \tag{7-71}$$

$$\hat{\hat{\dot{X}}}_{n+1} = \hat{\dot{X}}_{n+1} - \frac{(M-1)}{2}\ \Delta t\ (\hat{\hat{\ddot{X}}}_{n+1} + \hat{\ddot{X}}_{n+1}) \tag{7-72}$$

$$\hat{\hat{X}}_{n+1} = \hat{X}_{n+1} - \frac{(M-1)\Delta t}{2}\ (\hat{\hat{\dot{X}}}_{n+1} + \hat{\dot{X}}_{n+1}) - \frac{(M-1)^3}{12}\ (\Delta t)^3\ \hat{\ddot{X}}_{n+1} \tag{7-73}$$

for the third order QD filter.

The QD filter achieves smoothing effectiveness almost identical to that of the comparable constrained least squares filter without the usual array of saved input data samples. The QD is the real-time filter used in the majority of WSMR operational programs. References 42 through 45 discuss the QD filter in more detail.

APPENDIX A

1. Name or Acronym

Advanced Medium Range Air-to-Air Missile (AMRAAM) Parameter Estimation Program (APEP)

2. Contact

Mr. John E. Lindegren
3200 SPTW/KRTAR
Eglin AFB, FL 32542
(904) 882-4267
DSN: 872-4267

3. Documentation Sources

a. Bierman, G. J. Factorization Methods for Discrete Sequential Estimation. New York: Academic P, 1977.

b. Luenburger, D. G. Optimization by Vector Space Methods. New York: Wiley, 1969.

c. Sorenson, H. W. Kalman Filtering Techniques, Advances in Control Systems. Vol. 3, C. T. Leondes, ed. New York: Academic P, 1966.

4. Origin

Ball Systems Engineering (formerly VERAC, Incorporated)
10975 Torreyana Road, Suite 300
San Diego, CA 92121
14 February 1982

5. Comments

Strengths of the APEP filter are described in the following subparagraphs:

a. This filter provides optimally smoothed trajectory estimates in the same sense as an extended Kalman filter but with superior numerical accuracy because of echelonized square root information matrices with Householder orthogonal transformations of augmented state transition matrices. In addition, the matrices are partitioned for dynamic and measurement bias states and make explicit use of the block diagonal matrix structure to minimize computation steps for the sparse matrices.

b. The forward filter cycle is slightly slower than the standard Kalman filter, but the smoother cycle (backward filter cycle) is significantly faster than a standard fixed interval Kalman smoother. Smoothed estimates require only backward substitution, and smoothed covariances are generated with a UDU^T decomposition of the filter state covariances. Propagation of the smoothed square root information matrices does not require matrix inversions.

c. The procedure for "folding in" new measurements using Householder transformations to echelonize the measurement augmented state equations allows inclusion of asynchronous or irregular measurement data rates without reinitializing any of the filter solutions.

d. Trajectory propagation using the current filter estimates and state transition matrices avoids dependency on any particular measurement input.

e. All kinematic parameters are provided with 1-sigma error bounds for the estimates, which require propagation of the error covariances for indirect parameter derivations.

f. Data dropouts of short duration or changes in the number of available instruments are accommodated by APEP without catastrophic filter/smoother estimate errors or failures.

g. A review of filter cycle measurement residuals allows identification of wild points for edit and adjustment of a priori measurement covariances used in program input. (Automatic edit may be selected for a multiple of the expected residual standard deviation.) Deviations from the measurements error model used by the software are indicated by correlated measurement residuals.

Weaknesses of the APEP are described in the following subparagraphs:

a. If no inertial navigation system (INS) data are available for measurement input, the square root inverse filter (SRIF) cycle may have to be repeated to fine tune the a priori filter measurement and state noise covariances or to edit wild points which seriously perturb the estimates in the state propagation steps.

b. The smoothing cycle requires reversal of read order for the entire measurement interval, thus requiring all measurement data to be stored on disk files or other random access storage media. The algorithm uses fixed interval smoothing. The volume of data that can be processed at one continuous time interval is limited.

c. Data-processing time is significantly longer than the data-collection time interval if Time-Space-Position-Information (TSPI) from several instruments and from the inertial data are included as measurement inputs to the filter.

d. The present error model for TSPI instruments include only bias terms as the systematic error source. (The error model does provide for the Markov noise model.) For the short time intervals associated with air-to-air missile flights, bias terms may provide an adequate model. For longer time intervals, with large shifts in target range or angle relative to TSPI instruments, a more extensive error model would be required to ensure stationarity for the measurement residual.

FAST LEAST SQUARES

1. Name or Acronym

Least Squares Moving Arc Polynomial Using Recursive Sums

2. Contact

Mr. John E. Lindegren
3200 SPTW/KRTAR
Eglin AFB, FL 32542
(904) 882-4267
DSN: 872-4267

3. Documentation Sources

Sterrett, John K. "Manual for Moving Polynomial Arc Smoothing." Ballistic Research Laboratory Report 840. Aberdeen Proving Ground: Ballistic Research Laboratory, 1952.

4. Origin

Ms. Martha D. Everett
AFDTC/KRBA
Eglin AFB, FL 32542

5. Comments

The filter accomplishes its designed tasks very well. By changing the time span or the polynomial degree or both, the filter can be adapted to the trajectory, and by changing a single time constant, the polynomial can be evaluated at the end point for post-mission processing. No knowledge of measurement errors is required, and the wild point edit is adaptive because the distance off the curve considered defective is proportional to the noise in the data for the particular time span.

This filter has one weakness. It cannot handle a step function properly and thus filter gives poor results at missile booster ignition, burnout, or similar occurrences.

WEIGHTED LEAST SQUARES

1. **Name or Acronym**

   Weighted Least Squares Smoother

2. **Contact**

   Mr. Mike Dodgen
   Computer Sciences Corporation
   6545 Test Group/ENAC
   Hill AFB, UT 84056
   (801) 777-6497
   DSN: 458-6497

3. **Documentation Sources**

   Mr. Mike Dodgen
   Computer Sciences Corporation
   6545 Test Group/ENAC
   Hill AFB, UT 84056

4. **Origin**

   Kentron International, Incorporated

WLSRE in MITO26

1. Name or Acronym

Weighted Least Squares Recursive Estimation (WLSRE) contained in Multiple Instrument Trajectory Module (MITO26)

2. Contact

Mr. Robert Fierro
U.S. Army White Sands Missile Range
Attn: STEWS-NR-AM
White Sands Missile Range, NM 88002
(505) 678-2543
DSN: 258-2543

3. Documentation Sources

a. Fierro, Robert and Carolyn Nicholson. "Dynamic Optimized Smoothing Span (DYNOS)." Analysis and Computation Directorate Document. White Sands Missile Range: Analysis and Computation Directorate, 1970.

b. Comstock, D., M. Wright, and V. Tipton. "Handbook of Data Reduction Methods." Data Reduction Division Tech Rept. White Sands Missile Range: Data Reduction Division, 1964

c. Greene, Earl. "Edfil, A Routine for Editing and Filtering Data." Analysis and Computation Directorate Document. White Sands Missile Range: Analysis and Computation Directorate, 1977.

d. Comstock, D. "Introduction to Least Squares." Analysis and Computation Directorate Document. White Sands Missile Range: Analysis and Computation Directorate, 1968.

4. Origin

Software Branch
Data Sciences Division
Attn: NR-A
White Sands Missile Range, NM 88002
1970

5. Comments

The filter is embedded in several modules which make up the Modular Integrated Processing System (MIPS) of Data Sciences Division. The Multiple Instrument Trajectory Module (MITO26) exercises the filtering process most extensively. With this module,

a. observations are rotated to a common plane;

b. observations are filter-smoothed;

c. a set of predicted position, velocity, and acceleration components are obtained;

d. corrections for items such as refraction and velocity of propagation are made;

e. initial editing is performed to eliminate observations that have gross errors;

f. approximate position components are obtained using standard weights;

g. weights are determined for position-related observations;

h. best estimate of position components is obtained;

i. final editing of position-related observations is performed; (If any more position-related observations are rejected, steps f through h are repeated.)

j. approximate velocity components are obtained using standard weights;

k. weights are determined for velocity-related observations;

l. best estimate of velocity is obtained;

m. final editing of velocity-related observations are performed; (If any more velocity-related observations are rejected, steps j through l are repeated.)

n. approximate acceleration components are obtained using standard weights;

o. weights are determined for acceleration-related observations;

p. best estimate of acceleration components is obtained; and

q. final editing of acceleration-related observations is performed. (If any more acceleration-related observations are rejected, steps n through p are repeated.)

M-STATION

1. Name or Acronym

M-Station

2. Contact

Mr. Wen-Mi Liou
Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
(805) 989-7931
DSN: 351-7931

3. Documentation Sources

Liou, Wen-Mi. "Square Root Information Filter/Smoother for Multiple-Radar Tracking." Tech Note 3440-02-87. Point Mugu: Pacific Missile Test Center. March 1987.

4. Origin

Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
1987

5. Comments

The square root information filter/smoother had been designed to merge measurements from different radars to obtain the best estimate of trajectory. The measurements consist of range, azimuth, elevation, and range rate from radar tracking of up to 10 radars. This filter is a version of the conventional nine-state extended Kalman filter but has a fundamentally different approach to the optimal estimation problem. The whole updating process of the filter is founded on numerically stable, orthogonal transformation and preserves non-negativity of computed covariances. An ad hoc procedure for adaptive estimation had been implemented. In the implementation, the user has the options to select the time span for smoothing and to apply more weight on the measurements of some of the radars. Real radar tracking data were used to test the filter/smoother; the test results were satisfactory. In the testing, 100 data point smoothing was found adequate; consequently, more data point smoothing provided no significant improvement. This finding saves computer computation time and storage. As a result of this finding, a better divergence control method is being investigated.

PBS PROGRAM, M-STATION

1. **Name or Acronym**

Post Batch System (PBS) Program, M-Station

2. **Contact**

Mr. Wen-Mi Liou
Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
(805) 989-7931
DSN: 351-7931

3. **Documentation Sources**

Powers, W. "Multiple Station Radar Solution, Part I, Method of Calculation." Tech Note 3285-581. Point Mugu: Pacific Missile Test Center. September 1964.

4. **Origin**

Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
1964

5. **Comments**

Radar measurements are inversely proportional to estimates of their standard deviations. These weights are fixed and cannot change as the distance between target and site change; thus, separate passes are necessary during a tracking operation. The routine is fairly robust to individual site errors as each observation is edited if it exceeds a specified residual; however, to work well generally, individual radar data must be extensively pre-edited for on track (beacon) times, and bias errors should be corrected through calibration. The procedure produces a nine-state vector and its covariance matrix for each time point and, additionally can estimate individual radar biases.

PBS PROGRAM, DERIVE

1. Name or Acronym

Post Batch System (PBS) Program,DERIVE

2. Contact

Mr. Wen-Mi Liou
Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
(805) 989-7931
DSN: 351-7931

3. Documentation Sources

Morris, G. "Polynomial Smoothing and Differentiation by Least Squares." Working Note. Point Mugu: Pacific Missile Test Center, Code 3442, 1965.

4. Origin

Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
1965

5. Comments

This filter works well if data are well-approximated by a given quadratic expression. The technique may give poor estimation of end points where midpoint computations are not applied. The improved rate estimation is being examined through the PBS program, Optimum Finite Impulse Response Linear Phase Digital Filter (OPFILT).

PBS FILTER

1. **Name or Acronym**

Post Batch System (PBS) Filter

2. **Contact**

Mr. Wen-Mi Liou
Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
(805) 989-7931
DSN: 351-7931

3. **Documentation Sources**

Welch, M. Working Note 3442-22-79. Point Mugu: Pacific Missile Test Center, Code 3442, 1979.

4. **Origin**

Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
1965

5. **Comments**

This filter technique is superior to any time domain method for post-operations analysis; however, the results are poor when signal frequencies cannot be separated from noise frequencies or when the signal lies partly in transition zones. The usual end point problems occur because of the requirements for midpoint estimation. One solution to the end point problem is to collapse or expand the data at the ends, although the small filter size would result in a wide transition zone. Response function improvement is possible through a computer-aided design of filter weights. (See Post Batch System program Optimum Finite Impulse Response Linear Phase Digital Filter (OPFILT).)

PBS PROGRAM, OPFILT

1. Name or Acronym

Post Batch System (PBS) Program, Optimum Finite Impulse Response Linear Phase Digital Filter (OPFILT)

2. Contact

Mr. Wen-Mi Liou
Pacific Missile Test Center
Code 3400
Point Mugu, CA 93042
(805) 989-7931
DSN: 351-7931

3. Documentation Sources

a. Liou, Wen-Mi. "Optimum Finite Impulse Response (FIR) Linear Phase Digital Filters: Theory and Analysis of the McClellan-Parks Algorithm." Tech Note 3442-02-82. Point Mugu: Pacific Missile Test Center, May 1982.

b. McClellan, James A., Thomas W. Parks, and Lawrence R. Rabiner. "A Computer Program for Designing Optimum FIR Linear Filters." IEEE Trans. Audio Electroacoustics. AU-21.6 (1973): 506-26.

4. Comments

This general purpose filter design algorithm is capable of designing a large class of optimum (in the minimax sense) finite impulse linear phase digital filters such as low pass and high pass, as well as first and second differentiators. The algorithm can also be used to design filters which approximate arbitrary frequency specifications provided by the user. The user has control over the sizes of transition bands and ripples in both passband and stop band. The sharpening technique equipped with the filter design algorithm further reduces the sizes of small ripples in the designs of low-pass, high-pass, and band pass filters of any complexity. At the same time, this technique increases the sharpness of the frequency response in the transition bands.

ATAGAS KALMAN FILTER

1. **Name or Acronym**

Air-to-Air Gunnery Assessment System (ATAGAS) Kalman Filter

2. **Contact**

Mr. Lee Gardner
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-2628
DSN: 527-2628

3. **Documentation Sources**

Analytic Sciences Corporation. "Optimal Estimation for the Air-to-Air Gunnery Assessment System, Final Program Review." Reading: Analytic Sciences Corporation, n.d.

4. **Origin**

The Analytical Sciences Corporation
One Jacob Way
Reading, MA 01867
(617) 944-6850

5. **Comments**

The Analytical Sciences Corporation has a report which describes their evaluation of the filter's effectiveness. It was concluded that the filter's effectiveness was better but very costly computer time-wise.

MOTION TRACKING KALMAN FILTER

1. Name or Acronym

Motion Tracking Kalman Filter

2. Contact

Mr. Len Childers
6545 Test Group/ENAC
Hill AFB, UT 84056
(805) 777-8605
DSN: 458-8605

3. Documentation Sources

High Accuracy Multiple Object Tracking System (HAMOTS) Computer Software Maintenance Manual, December 1979.

4. Origin

Messrs. Doug Troxler and Mick Chaplin
General Dynamics
Electronics Division
San Diego, California

5. Comments

The filter does not handle wild points from input data as well as expected.

TEC TRACKER

1. Name or Acronym

U.S. Army TEXCOM Experimentation Center (TEC) Tracker (formerly Combat Developments Experimentation Center (CDEC) Kalman Filter)

2. Contact

Dr. Joseph M. Weinstein
Scientific Support Laboratory
P.O. Box 100
Fort Ord, CA 93491
(408) 384-2161

Ms. Julie Lemen
Scientific Support Laboratory
P.O. Box 898
Jolon, CA 93928
(408) 385-2880

3. Documentation Sources

a. Weinstein, Joseph M. "Position Location Logic." TEC Scientific Support Laboratory Document. Jolcn: TEC Scientific Support Laboratory, June 1988.

b. Weinstein, Joseph and others. "Kalman Filter User Manual." Revision I. CDEC Scientific Support Laboratory Document. Jolon: CDEC Scientific Support Laboratory, July 1981.

4. Origin

Kalman tracking software developed by General Dynamics staff in 1971 was furnished to TEC for use with Range Measuring System (RMS) hardware. The TEC has since upgraded tracking procedures, logic, and code, notably around 1976 and 1980 but especially since 1987.

5. Comments

Description. The TEC Tracker provides three-dimensional position and velocity of ground and air players during instrumented field combat simulations. This tracker is comprised of a Kalman filter and several other real-time routines which together process ranges between A and B units (RMS ranging and player units) as well as altimeter and other input. Post-operation routines smooth and analyze tracking output and apply alternative tracking algorithms to logged real-time input.

Performance. The tracker meets TEC's main real-time need for tracking as many as 100 players reasonably well with 10 meters horizontal error. As upgraded, the tracker reinitializes quickly as needed, does not require time-clumped ranges to any given B-unit, and prefilters (with outlier rejection) ranges between any given A,B pair before using them in the Kalman filter.

Prospects. The Kalman computations are intensive and must rely on predictive modeling of ground and air combat maneuver dynamics. Recent study suggests that throughput and accuracy may gain from replacing the Kalman filter by a simpler routine which relies just on good quality measured inputs, for example, a linearized least squares fit to the prefiltered ranges.

DRIFT TRACKING KALMAN FILTER

1. Name or Acronym

Drift Tracking Kalman Filter

2. Contact

Mr. Len Childers
6545 Test Group/ENAC
Hill AFB, UT 84056
(801) 777-8605
DSN: 458-8605

3. Documentation Sources

High Accuracy Multiple Object Tracking System (HAMOTS) Computer Software Maintenance Manual. December 1979.

4. Origin

General Dynamics
Electronics Division
San Diego, California
1979

5. Comments

This filter has not been used because score pod equipped vehicles are not available at Utah Test and Training Range.

EATS KALMAN FILTER

1. **Name or Acronym**

Extended Area Test System (EATS) Kalman Filter

2. **Contact**

Ms. L. Wilson
Pacific Missile Test Center
Code 3452
Point Mugu, CA 93042
(805) 984-8784
DSN: 351-8784

3. **Documentation Sources**

The General Dynamics, Electronics Division, System Controller Design document for EATS.

Any technical reference for Kalman filtering.

4. **Origin**

General Dynamics
Electronics Division
San Diego, California
1978 to 1980

5. **Comments**

The EATS design requirements called for real-time state vector estimation with accuracy constraints. This six-state Kalman design provided the best alternative while satisfying design goals. Real-time accuracies can be improved with finer tuning capability which is currently available with tri-level (low-medium-high) dynamic tuning parameter. This dynamic tuning parameter appears satisfactory for most purposes. Post-operations accuracy improvement techniques are currently under study.

TRIDENT KALMAN FILTER

1. Name or Acronym

Trident Missile Tracking Kalman Filter

2. Contact

Mr. Donald Olson
Pacific Missile Test Center, Code 1051
Point Mugu, CA 93042-5000
(805) 989-8804
DSN: 351-8804

Mr. Eric Senor
Pacific Missile Test Center, Code 3442
Point Mugu, CA 93042-5000
(805) 989-7931
DSN: 351-7931

3. Documentation Sources

a. Gelb, Arthur. Applied Optimal Estimation. Cambridge: MIT P, 1974.

b. Olson, Donald. "Filter and Smoother for Trident Missile Tracking." TP-64. Point Mugu: Pacific Missile Test Center, June 1988.

4. Origin

Designed by Pacific Missile Test Center personnel, the preliminary covariance simulations of the Trident Kalman filter began in 1977. These simulations led to the first operational version which supported four missile tests in the Pacific during 1983-1984. In 1984, a major redesign effort resulted in a more reliable and accurate second version which has supported 13 operational tests thus far.

5. Comments

For range safety redundancy, the nine-state extended Kalman filter operates on two CDC Cyber 175 mainframes. Square-root filtering was found to offer no advantage over standard covariance filtering in this application, presumably because of the large, 60-bit word size. Measurements consist of range sums and range rate sums from five transmitting stations operating 20 milliseconds apart. In addition, pedestal angles at the receiving telemetry antenna are used. While process noise is constant, the filter switches to a lower value at third stage burnout. Constants are also used for measurement noise standard deviations and edit limits.

Accurate initialization was found to be crucial to filter performance throughout missile flight. The initial state estimate is computed in a tangent plane, rectangular coordinate system. Ranging data from three stations are filtered with a recursive, first-degree polynomial filter, then used to solve for missile position and velocity. Because the z component of the ranging solution is inaccurate at low elevations, z position and velocity are obtained from missile telemetry. If telemetry is not available, the scheme defaults to a combination of nominal data and the ranging solution.

QD - WSMR

1. **Name or Acronym**

Quadratic Digital (QD) Filter

2. **Contact**

Mr. John Falke
U.S. Army White Sands Missile Range
Attn: STEWS-NR-AR
White Sands Missile Range, NM 88002
(505) 678-3458
DSN: 258-3458

3. **Documentation Sources**

McCool, W. A. "QD-A New Efficient Digital Filter." Analysis and Computation Directorate Internal Memorandum 60. White Sands Missile Range: Analysis and Computation Directorate, August 1967.

4. **Origin**

Mr. W. A. McCool
Acting Director
Analysis and Computation Directorate
White Sands Missile Range, NM 88002
August 1977

5. **Comments**

The quadratic digital filter's computing time is extremely small and invariant with point span. It achieves smoothing effectiveness almost identical to that of the comparable constrained least-squares filter.

QD - EGLIN

1. Name or Acronym

   Quadratic Digital (QD) Filter

2. Contact

Mr. John E. Lindegren
3200 SPTW/KRTAR
Eglin AFB, FL 32542
(904) 882-4267
DSN: 872-4267

3. Origin

   Eglin AFB, Florida

4. Comments

Although this filter is efficient, extremely fast, easy to use, and does not require extensive data arrays, its instabilities caused by unedited wild points limits its use.

QD - APO SAN FRANCISCO

1. **Name or Acronym**

Quadratic Digital (QD) Filter

2. **Contact**

Mr. Sonny Padayhag
Kentron International, Incorporated
Box 1207
APO San Francisco, CA 96555
(805) 238-7994, Ext. 8-2020
DSN:254-2020

3. **Documentation Sources**

Real-time program (RTP) or real-time impulse filter (RTIF) documentation.

4. **Comments**

This filter works well for speedy execution and extrapolation of small periods of time such as 200 milliseconds. It does not perform well with exoatmospheric, ballistic trajectories and long extrapolations or propagations.

QD - YUMA

1. Name or Acronym

Quadratic Digital (QD) Filter

2. Contact

Mr. Robert Mai
Yuma Proving Ground
Attn: STEYP-MT-TA
Yuma, AZ 85365
(602) 328-3295
DSN: 899-3295

3. Documentation Sources

a. McCool, W. A. "QD-A New Efficient Digital Filter." Analysis and Computation Directorate Internal Memorandum 60. White Sands Missile Range: Analysis and Computation Directorate, August 1967.

b. Mai, Robert W. "The QD Filter in YPG's Real-Time Laser Display System." STEYP-MAC Document. Yuma: Yuma Proving Ground.

4. Origin

Mr. Robert Mai
Yuma Proving Ground
Attn: STEYP-MAC
Yuma, AZ 85365-9102
1974

5. Comments

The quadratic digital filter does a much better job of estimating velocity than conventional least squares moving arc routines. Computation time is very small and does not depend on filter memory length. This easy-to-use filter is very simple to put into computer code. Filter response to impulse changes in acceleration is slow, that is, approximately two to three times the memory length selected, so filter estimates are useful primarily during segments where acceleration is minimal. The filter memory length, which determines the constant gain coefficients for position, velocity, and acceleration, can be adjusted to accommodate higher accelerations but then the noise on the filter estimates increases. Perhaps the major drawback of the QD filter is the difficulty in determining accuracy of velocity estimates which is dependent upon the complex interaction of sampling rate, target, dynamics, filter memory length, and noise on the measurements.

QUADRATIC POLYNOMIAL FILTER

1. **Name or Acronym**

Quadratic Polynomial Filter

2. **Contact**

Dr. Floyd Hall
Naval Weapons Center
Code 62303
China Lake, CA 93555
(619) 939-6346
DSN: 437-6346

3. **Documentation Sources**

Gossett, Eric. "On-Axis Tracking System."

4. **Comments**

This filter works quite well for real-time trajectory data from radar or laser data. The quadratic polynomial filter is an α-β-γ filter based on two parameters. This filter is a little different from what is expressed in Gossett's paper although it gives the same results. The filter is not considered adequate for close-in targets with tracking radar problems. Naval Weapons Center is currently developing a filter that rectifies this inadequacy. The new filter will have nonlinear constraints.

QD - NATC

1. Name or Acronym

Quadratic Digital (QD) Filter

2. Contact

Mr. John Shields
Naval Air Test Center
Computer Sciences Division (CS35)
Patuxent River, MD 20670-5304
(301) 863-3396
DSN: 326-3396

3. Documentation Sources

McCool, W. A. "QD-A New Efficient Digital Filter." Analysis and Computation Directorate Internal Memorandum 60. White Sands Missile Range: Analysis and Computation Directorate, August 1967.

4. Origin

Mr. Frederick K. H. Hoeck
Computer Services Directorate
Patuxent River, MD 20670-5304
1979

5. Comments

This filter is adequate for the application; however, as with all filters, there is some setting time. Comparisons made to other data sources were favorable. The filter requires little memory because samples need not be stored. Constants need to be calculated based on sample rate and number of points in the point span. It handles data errors and recovers with few problems.

AVRAGE

1. **Name or Acronym**

AVRAGE

2. **Contact**

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-0871
DSN: 527-0871

3. **Documentation Sources**

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Average Data Smoothing Subroutine." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Chapter 8. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. **Origin**

Systems Development Corporation
June 1972

5. **Comments**

The AVRAGE gives an average of up to 61 consecutive input data values.

1. Name or Acronym

Differentiation by Least Squares Subroutine (DUZ2)

2. Contact

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB, CA 93523
(805) 277-0871
DSN: 527-0871

Mr. Mike Tietz
Computer Sciences Corporation
P.O. Box 446
Edwards AFB, CA 93523
(805) 277-3800
DSN: 527-3800

3. Documentation Sources

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Differentiation by Least Squares Subroutine (DUZ2)." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Chapter 20. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. Comments

The DUZ2 calculates smooth values as well as first and second derivatives by fitting a least squares parabola. This subroutine works well with low-angular acceleration but not with high-angular accelerations. It is limited in how much data can be processed in one call to the subroutine by the array sizes, because an excessive amount of computer memory could be required.

SDIFFR

1. **Name or Acronym**

Differentiation Subroutine (SDIFFR)

2. **Contact**

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-0871
DSN: 527-0871

3. **Documentation Sources**

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Differentiation Subroutine (SDIFFR)." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Chapter 20. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. **Origin**

Messrs. R. C. Schramand R. T.Scott
White Sands Missile Range, New Mexico
July 1972

5. **Comments**

Basically, the SDIFFR functions as a single-parameter, cycling DIRSIT which creates parameter arrays containing up to 60 time points of smooth values, first derivatives, and second derivatives as desired. The SDIFFR is slightly different from DIRSIT in that it is called once for each desired parameter, initializing (if requested), and processing up to 60 points and then returns. Thus, SDIFFR cycles up to 60 points inside itself for 1 parameter instead of being called up to 60 times.

As a method of computation, the initialization is accomplished by fitting a least squares parabola and using a maximum likelihood procedure to the first user-designated interval input data points. The DIRSIT process then takes over for subsequent points. Because this process is not self-starting, at least the first interval of points should be allowed at the beginning for start-up. Once the DIRSIT process begins, a second-order Taylor series expansion is adjusted according to certain statistical criteria until the number of points above the curve differs from the number below the curve by a user-designated amount.

The first and second derivatives of the resulting second-order curve are taken as the derivatives at the first of the interval. The interval is then moved by one point, and the process continued until the end of data is reached. At the end of the data, the interval size is collapsed until only one point remains. The derivatives for the last point are set equal to the derivative of the next to the last point. Because of this interval collapsing, up to INPTS-1 points should be disregarded at the end of the data, where INPTS is the number of points in the interval.

A wild point has only minimal effect on the DIRSIT process since the only test made is whether it is above or below the curve. How far above or below is of little importance.

The conditions of validity are listed below.

a. Care should be taken to avoid processing large discontinuities; however, time increments need not necessarily be constant.

b. The appropriate slope parameter and tolerance criteria parameter must be provided to ensure valid results.

c. The number of input data points per parameter is limited to 60 for each call. More points can be accommodated through multiple calls to SDIFFR.

d. The SDIFFR is not a self-starting routine. The initialization process requires a number of points to get things moving, so the first several time points of derived values will be poor approximations.

e. The SDIFFR uses a collapsing technique to process the last interval of time points, but the results for these last several points (INPTS-1) become increasingly degraded as SDIFFR is forced to work on smaller and smaller intervals.

f. The calling program for SDIFFR uses an overlapping technique to avoid including the invalid results at the beginning and end of each array processed. A check should be made to ensure that the overlap is adequate so that no major discontinuities appear at each overlap point.

g. The calling program for SDIFFR uses a default interval size which the user may change.

DIRSIT

1. **Name or Acronym**

   Derivative Information Recovery by a Selective Integration Technique (DIRSIT)

2. **Contact**

   Mr. Len Childers
   6545 Test Group/ENAC
   Bldg 1284
   Hill AFB, UT 84056
   (801) 777-8605
   DSN: 458-8605

3. **Documentation Sources**

   Utah Test and Training Range postflight data reduction documents.

4. **Origin**

   Mr. Len Childers
   6545 Test Group/ENAC
   Hill AFB, UT 84056-5000
   (Adaptation of SDIFFR)

UFTAS FILTER OVERLAY

1. Name or Acronym

Uniform Flight Test Analysis System (UFTAS) Digital Filtering Primary Overlay (FILTER)

2. Contact

Dr. William G. Kitto
6521 Range Squadron/RCP
EdwardsAFB, CA 93523
(805) 277-3198
DSN: 527-3198

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-0871
DSN: 527-0871

3. Documentation Sources

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "UFTAS Digital Filtering Primary Overlay (FILTER)." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. Origin

Dr. William G. Kitto
1982

5. Comments

Currently, this filter is used heavily in the Frequency Response Analysis (FRA) program. (Butterworth or User Supplied Weight Type filter.)

BET

1. **Name or Acronym**

Best Estimate of Trajectory (BET)

2. **Contact**

Mr. Robert Mai
Yuma Proving Ground
Attn: STEYP-MT-TA
Yuma, AZ 85365
(602) 328-3295
DSN: 899-3296

3. **Documentation Sources**

Liebelt, Paul B. An Introduction to Optimal Estimation. Reading: Addison-Wesley, 1967.

4. **Origin**

Mr. Robert Mai
Yuma Proving Ground
Attn: STEYP-MAC
Yuma, AZ 85365
1976

5. **Comments**

The Kalman filter and Rauch-Tung-Striebel smoother take range, range rate, azimuth, elevation, and apparent angle measurements from up to 20 instruments and based upon a priori statistical information about the trajectory, the measurements, and the initial conditions, optimal estimates of the state vector are computed at each of the measurement times. The state vector size is expandable from the basic three components (x,y,z coordinate system) to include up to five derivatives: measurement source, pitch, yaw, pitch rate, and yaw rate. Estimates of the state vector are optimal in the sense that the uncertainty in each component of the state vector is minimized. This BET can provide excellent results; however, the results are only as good as the a priori statistical information. For applications where knowledge of measurement uncertainty and dynamics in the trajectory are well known or easily estimated, then the BET will undoubtedly provide very close to the absolute best estimate. If the a priori statistics distort the truth, then the BET trajectory will be distorted. In actual practice at Yuma Proving Ground, a BET trajectory is derived in an interactive fashion. Initial runs are made with a priori statistics implying more uncertainty than is actually present. Then analysis of the measurement residuals and estimated dynamics are used to refine the a priori estimates. A new BET trajectory is then generated and analysis of residuals and dynamics undertaken. This process continues until the a priori statistics match the derived estimates.

There are significant disadvantages to using the BET. Although the program setup is very complex and tedious, obtaining *a priori* statistical information is often perceived as an art rather than a science. Errors in setup or in judgment almost always have a detrimental effect on the final estimates. In addition, computation time is very long, and the required iterative process can cause data turnaround time to expand to longer than two weeks. So if you want the very best, you must be willing to pay for it!

RTUF

1. Name or Acronym

Real-Time Update Filter (RTUF)

2. Contact

Mr. Andy Roy or Joe Warren
Computer Sciences Raytheon
Unit 2310
Eastern Space and Missile Center
Patrick AFB, FL 32925
(407) 494-7133
DSN: 854-7133

Mr. J. V. Copp
Computer Sciences Raytheon
Eastern Space and Missile Center
Patrick AFB, FL 32925
(407) 853-7783

3. Origin

Ms. Marie Colmer and Mr. Royal Pepple
RCA/MTP
Eastern Space and Missile Center
Patrick AFB, FL 32925

1971

4. Comments

This filter is used mainly for the Missile Precision Instruction Radar (MIPIR) class of radars. These radars are large scale. The RTUF is recursive, adaptive, exponentially weighted, and has fading memory. By adaptive, it is meant that the filter can adjust as the data gets seemingly noisier or less noisy. As the data becomes noisier, the bandwidth is decreased, and as the data becomes less noisy the bandwidth increases. In an adaptive filter, the coefficients of the terms involving the difference between the predicted value and the given raw value vary with time. In an exponentially weighted filter such as this one, the coefficients have the time variable as part of their exponents. In a fading-memory filter, the memory fades with time. In this case, the memory fades exponentially. In other words, newer data are weighted considerably more than older data when predicting a new value.

The RTUF is a simple filter because it does not work with partial derivatives or matrices. For this reason, Mr. Joe Warren says that this filter was good in the 1960s when computers were slower and had less memory. Mr. Warren predicts that no more RTUFs will be used within two to three years and suggests using a Kalman-type filter in lieu of this one.

RFILTR

1. Name or Acronym

Real-Time Software System (RFILTR)

2. Contact

Mr. Robert Crolene
Pacific Missile Test Center (formerly Pacific Missile Range (PMR))
Code 1074
Point Mugu, CA 93042
(805) 982-8073
DSN: 351-8073

3. Documentation Sources

Cragun, G. C. "Real-Time Data Filtering." PMR Tech Note 3285-576. Point Mugu: Pacific Missile Range.

4. Origin

NAMTC
Code 3400
1965

5. Comments

The RFILTR is a simple, exponential adaptive filter. The advantage of this filter is its simplicity. Upgrading can be achieved with some adaptive smoothing controls by basing it on residuals or by employing a Kalman algorithm. However, this step would complicate filter operations and may not be practical for the software system of the Univac 1230 for which this filter was originally designed. Upgrading the filter for use in a mainframe computer should present no problems.

Major problems with this filter are smoothing control and initialization variance control. Correcting one control usually makes the other one worse. Initialization errors are especially large for velocities and accelerations.

RCHECK

1. **Name or Acronym**

Rate Check Data Smoothing Subroutine (RCHECK)

2. **Contact**

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-0871
DSN: 527-0871

3. **Documentation Sources**

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Rate Check Data Smoothing Subroutine (RCHECK)." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Chapter 8. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. **Origin**

Mr. G. A. Lott
Lockheed-Georgia Company
June 1972

The RCHECK was successfully used by the Lockheed-Georgia Company in processing C-5A test data and was later modified to fit into UFTAS.

5. **Comments**

This subroutine searches for wild points by comparing DLY, the absolute difference between the current and previous point values, with TOL, a linear function of the average of successive differences within a given interval. The TOL represents the tolerance limit for the interval's data variations. The multiplicative and additive factors of this linear function are user-supplied. The TOL is increased or decreased from point to point according as DLY is greater than or less than TOL. In addition, if DLY is greater than or equal to TOL, the current data value is replaced by the last valid one.

EYBALL

1. Name or Acronym

Eyball Data Smoothing Subroutine (EYBALL)

2. Contact

Mr. Ly V. Tran
6521 Range Squadron/RCP
Edwards AFB CA 93523-5000
(805) 277-0871
DSN: 527-0871

3. Documentation Sources

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Eyball Data Smoothing Subroutine (EYBALL)." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. Origin

Mr. C. F. Carpenter
General Dynamics

Capt J. H. Pierson, USAF
Edwards AFB, CA
July 1972

5. Comments

The user specifies a value DELTA that represents the difference between the maximum and minimum values for a given parameter. The value of DELTA is based on the user's prior knowledge of what values the parameter should take.

If the actual range of values within the interval is less than or equal to DELTA, then the output parameter value is the average of the values in the interval. If the range is greater than DELTA, then the weighted average of the most common values in the interval, with the more central values given the greater weight, is the output parameter value.

The advantages of the EYBALL subroutine are

a. the EYBALL closely approximates intuitive treatment of data,

b. the user has control by using prior knowledge of the parameter,

c. the wild points are discarded before determination of the output parameter value, and

d. the excessive wild points are identified.

ON-AXIS

1. **Name or Acronym**

On-Axis Radar Target Tracking System (ON-AXIS)

2. **Contact**

Mr. John E. Lindegren
3200 SPTW/KRTAR
Eglin AFB, FL 32542
(904) 882-4267
DSN: 872-4267

3. **Documentation Sources**

"On-Axis: Philosophy/Technology/Development." Advanced Research Project Agency, Radar Microwave Link Tech Memo 211. Patrick AFB: Advanced Research Project Agency, Radar Microwave Link, 15 December 1970.

4. **Origin**

Advanced Research Project Agency
Radar Microwave Link
Patrick AFB, Florida
1969

Modified: Air Force Development Test Center
AFDTC/KR
Eglin AFB, Florida
1973

5. **Comments**

Strengths of this filter include self-calibration capability through stellar track; adaptive track gain with smooth, low-noise track; and predetermined missile trajectory slew at launch.

This filter's weaknesses consist of requiring a well-qualified crew and careful maintenance and software control of the computer system and data base. The On-Axis track accuracy (with pedestal position pick-offs) offers no significant improvement over normal

AN/FPS-16 TSPI for aerodynamic maneuvering targets if standard post-mission smoothing procedures (moving-arc-polynomial midpoint fits to raw data) are used.

DYNO43

1. Name or Acronym

DYNO43

2. Contact

Mr. Robert Fierro
U.S. Army White Sands Missile Range
Attn: STEWS-NR-AM
White Sands Missile Range, NM 88002
(505) 678-2543
DSN: 258-2543

3. Documentation Sources

a. Comstock, D., M. Wright, and V. Tipton. "Handbook of Data Reduction Methods." Data Reduction Division Tech Rept. White Sands Missile Range: Data Reduction Division, 13 August 1964.

b. Comstock D. Introduction to Least Squares. White Sands Missile Range: Analysis and Computation Directorate, 1968.

4. Origin

Classical Least Squares Smoothing
August 1970

5. Comments

The filter derives smooth data from observation using the least squares moving arc method. The smoothed positions are differentiated to obtain velocities which, in turn, are differentiated to obtain accelerations.

GNFL

1. **Name or Acronym**

   Gennery Filter (GNFL)

2. **Contact**

   Dr. Kenneth Lane
   CSR 3200
   P.O. Box 4127
   Patrick AFB, FL 32925

3. **Documentation Sources**

   a. Computer Program 285, GNFL

   b. Gennery, Donald B. "An Improved Digital Filter." Mathematical Services TM-63-8, Dec. 1963.

4. **Origin**

   Mr. Donald B. Gennery
   1963

5. **Comments**

This filter is designed to give a frequency response with sharper roll-off and with increased attenuation at higher frequencies when compared to standard least squares polynomial filters.

1. Name or Acronym

Gennery Smoother (GNSM)

2. Contact

Dr. Kenneth Lane
CSR 3200
P.O. Box 4127
Patrick AFB, FL 32925

3. Documentation Sources

a. Computer Program 586, GNSM

b. Gennery, Donald B. "Direct Digital Filters for General Purpose Use." RCA Document. Patrick AFB: RCA/MTP, January 1966.

4. Origin

Mr. Donald B. Gennery
1966

5. Comments

Designed as an improvement of the GNFL filter with sharper roll-off, this filter has the additional feature of being able to bridge discontinuities in the derivatives of the input data.

FIRFILT

1. **Name or Acronym**

Finite Impulse Response Filter (FIRFILT)

2. **Contact**

Mr. Jerry Biedscheid
Sandia National Laboratories
Division 7522
P.O. Box 5800
Albuquerque, NM 87185
(505) 844-4048
DSN: 244-4048

3. **Documentation Sources**

McClellan, James A., Thomas W. Parks, and Lawrence R. Rabiner. "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters." IEEE Trans. Audio Electroacoust. AU-21.6 (1973): 506-26.

4. **Origin**

Mr. D. J. Miller
Sandia National Laboratories
Division 1414
P.O. Box 5800
Albuquerque, NM 87185
1980

5. **Comments**

This filter gives good results in low dynamic situations, but the filter is difficult to control.

RECFLT

1. Name or Acronym

RECFLT

2. Contact

Mr. W. D.Swartz
Sandia National Laboratories
Division 7524
P.O. Box 5800
Albuquerque, NM 87185
(505) 844-2237

3. Documentation Sources

Stearns, S. D. Digital Signal Analysis. Rochelle Park: Hayden, 1975.

4. Origin

Mr. D. J.Miller
Sandia National Laboratories
Division 7524
P.O. Box 5800
Albuquerque, NM 87185
1980

5. Comments

This filter provides excellent results, particularly with large data sets using the phase shift removal option.

FILMAX

1. **Name or Acronym**

   FILMAX

2. **Contact**

   Mr. D. O. Smallwood
   Sandia National Laboratories
   Division 7544
   P.O. Box 5800
   Albuquerque, NM 87185
   (505) 844-1074
   DSN: 244-1074

3. **Documentation Sources**

   Smallwood, D. O. "An Improved Recursive Formula for Calculating Shock Response Spectra." Shock and Vibration Bulletin. 51.2 (1981): 211-17.

4. **Origin**

   Mr. D. O. Smallwood
   Sandia National Laboratories
   Division 7544
   1979

5. **Comments**

The older recursive models of this filter used for calculating the shock response spectra resulted in significant errors when the natural frequency exceeded one-sixth of the sample rate. This new filter avoids the problem.

1. Name or Acronym

   MDPT51

2. Contact

   Mr. J. A. Ward
   Eastern Space and Missile Center
   RCA/MTP
   Patrick AFB, FL 32935

3. Documentation Sources

   a. RAID Computer Program

   b. "Filtering Tracking Data for Range Safety Displays." RCA Document. Patrick AFB: RCA/MTP, January 1984.

1. **Name or Acronym**

BSMW

2. **Contact**

Mr. J. A. Ward
Computer Sciences Raytheon
Eastern Space and Missile Center
Patrick AFB, FL 32925

3. **Documentation Sources**

a. RAID Computer Program

b. "Filtering Tracking Data for Range Safety Displays." RCA Document. Patrick AFB: RCA/MTP, January 1984.

4. **Comments**

The BSMW is a quadratic filter that is recursive but not adaptive. (A nonadaptive filter is a filter with a fixed bandwidth. The coefficients of the terms involving the difference between the predicted value and the given raw value are constant; they do not vary with the perceived noise content of the data.)

1. Name or Acronym

SMW

2. Contact

Mr. J. A. Ward
Computer Sciences Raytheon
Eastern Space and Missile Center
Patrick AFB, FL 32925

3. Documentation Sources

a. RAID Computer Program

b. "Filtering Tracking Data for Range Safety Displays." RCA Document. Patrick AFB: RCA/MTP, January 1984.

4. Comments

The SMW is a linear filter that is recursive but not adaptive. (A nonadaptive filter is a filter with a fixed bandwidth. The coefficients of the terms involving the difference between the predicted value and the given raw value are constant; they do not vary with the perceived noise content of the data.)

FREQUENCY RESPONSE ANALYSIS

1. **Name or Acronym**

Frequency Response Analysis (FRA)

2. **Contact**

Mr. Barry Mishler
6521 Range Squadron/RCP
Edwards, CA 93523-5000
(805) 277-6040
DSN: 527-6040

3. **Documentation Sources**

Computer Sciences Branch, 6521 Range Squadron, 6510 Test Wing. "Frequency Response Analysis (FRA) Overlay." Uniform Flight Test Analysis System (UFTAS) Reference Manual. Version 3.1. Edwards Air Force Base: Air Force Flight Test Center, October 1990.

4. **Origin**

Mr. Tom Twisdale
6510 Test Wing/DOEF
Edwards AFB, CA 93523-5000
(805) 277-1248
DSN: 527-1248

Dr. William G. Kitto
6521 Range Squadron/RCP
Edwards AFB, CA 93523-5000
(805) 277-3198
DSN: 527-3198

5. **Comments**

The Frequency Response Analysis program transfers the dynamic time domain data into the frequency domain to do the frequency response analysis specified by the user. Power spectral densities, transfer functions, and coherence functions can be calculated in the frequency domain. The dynamic time domain data are contained on either a B-file or a C-file. After the desired analysis is complete, FRA can put the frequency domain results back into the time domain.

The FRA program is not a filter. This program is meant to aid the user in determining what filter to use and how to use it. For instance, the FRA can aid in determining what cutoff frequencies should be in the frequency spectrum or if the filter should be bandlimited or nonbandlimited. At the present time, the program cannot accept more than 1024 time points. It uses the first 1024 time points received and discards the rest of the points.

REFERENCES

1. Bierman, Gerald J. Factorization Methods for Discrete Sequential Estimation. New York: Academic P, 1977.

2. Gelb, Arthur. Applied Optimal Estimation. Cambridge: MIT P, 1974.

3. Maybeck, Peter S. Stochastic Models, Estimation and Control. 3 vols. New York: Academic P, 1979. Vol. 3.

4. Morrison, Norman. Introduction to Sequential Smoothing and Prediction. New York: McGraw-Hill, 1969.

5. Bucy, Richard S., and Peter D. Joseph. Filtering for Stochastic Processes with Applications to Guidance. New York: Interscience Publishers, 1968.

6. Data Reduction and Computing Group. Error Analysis and Methods for Estimating Errors in Position, Velocity, and Acceleration Data. Document 153-71. White Sands Missile Range: Range Commanders Council Secretariat, 1971.

7. Gennery, Donald B. "Direct Digital Filters for General Purpose Use." RCA Document. Patrick Air Force Base: RCA/MTP, 1966.

8. Berry, Floyd N. "QD Filters Compared with α-β-γ Filters." Minutes of 35th Data Reduction and Computing Working Group Meeting. 19-23 Oct 1970. White Sands Missile Range: Range Commanders Council Secretariat, 1970. 132-42.

9. Leondes, Cornelius T. Theory and Applications of Kalman Filtering. N.p.: Harford House, Feb. 1970.

10. Moore, Alan J. "A Study of the Optimum Techniques for Radar and Doppler Tracking Data Digital Filtering." Systems Development Corporation Document. Santa Monica: Systems Development Corporation, 1969.

11. Ormsby, Joseph F. A. "Design of Numerical Filters with Applications to Missile Data Processing." Space Technology Laboratories Document. Los Angeles: Space Technology Laboratories, 1960.

12. Smith, E. S. "A Collection of Linear Numerical Filters for Missile Trajectory Data with General Principles and Design Considerations." RCA Document. Patrick Air Force Base: RCA, 1966.

13. Stirton, R. J. "Discrete Numerical Filters." U.S. Naval Ordnance Test Station Document. China Lake: U.S. Naval Ordnance Test Station, 1966.

14. Taylor, James T. "Digital Filters for Non Real Time Data Processing." NASA Document. Huntsville: NASA, George C. Marshall Space Flight Center, 1967.

15. McCool, W. A. "Derivation of the Discrete Kalman Filter Formulation Based on the QD Filter and Classical Least Squares Theories." Analysis and Computation Directorate Tech Rept 9. White Sands Missile Range: Analysis and Computation Directorate, 1969.

16. Hansen, Allen J. "624A Velocity Filter." Federal Electric Corporation Document. Vandenberg Air Force Base: Federal Electric Corporation, 1965.

17. Siler, Henry K. "Procedure for Computation of Frequency Constrained Digital Filter Coefficients." RCA Document. Patrick Air Force Base: RCA, 1963.

18. Pauley, Richard F. "A Unified Approach to Data Smoothing." RCA Document MTC-TDR-10. Patrick Air Force Base: RCA, 1962.

19. Cox, Henry. "Recursive Nonlinear Filtering." David Taylor Model Basin. Washington, DC: n.p., n.d.

20. McCool, W. A. "Techniques for Comparing the Performance of General-Purpose Digital Filters." Analysis and Computation Directorate Internal Memorandum 63. White Sands Missile Range: Analysis and Computation Directorate, 1967.

21. Rader, Charles M. "Digital Filter Design Techniques." MIT Lincoln Laboratory Tech Note 1965-63. Lexington: Lincoln Laboratory, 1965. [AD 627146].

22. Ford, Wayne T., and James H. Hearne. "Least Squares Inverse Filtering." Geophysics. 31.5 (1966): 917-26.

23. Berry, Floyd N. "Evaluation of Several Recursive Random Error Estimation Techniques." Minutes of 35th Data Reduction and Computing Working Group Meeting. 19-23 Oct 1970. White Sands Missile Range: Range Commanders Council Secretariat, 1970. 143-62.

24. Holtz, Howard, and C. T. Leondes. "The Synthesis of Recursive Digital Filters." J. Assoc. Comput. Mach. 13.2 (1966): 262-80.

25. Smith, F. T. The Application of the Kalman Filter to Some Astrodynamical Problems. Rand Corporation Memorandum RM-4277-PR. Santa Monica: Rand Corporation, 1964.

26. Kaplan, Wilfred. Operational Methods for Linear Systems. Reading: Addison-Wesley, 1962. 391-95.

27. Quenouille, M. H. The Analysis of Multiple Time-Series. New York: Hafner, 1957. 65-81.

28. Lueg, Russell E. Basic Electronics for Engineers. Scranton: International Textbook, 1963. 140-45.

29. Bendat, Julius S., and Allan G. Piersal. Measurement and Analysis of Random Data. New York: Wiley, 1966.

30. D'Esopo, D. A. "A Note on Forecasting by the Exponential Smoothing Operator." Oper. Res. 9.5 (1961): 686-87.

31. Brown,Robert G., and Richard F. Meyer. "The Fundamental Theorem of Exponential Smoothing." Oper. Res. 9.5 (1961): 673-85.

32. Conte, Samuel D. Elementary Numerical Analysis: An Algorithmic Approach. New York: McGraw-Hill, 1965.

33. Hamming, Richard W. Numerical Methods for Scientists and Engineers. New York: McGraw-Hill, 1962.

34. Drylov, V. I. Approximate Calculation of Integrals. New York: MacMillan, 1962.

35. Ralston, Anthony. A First Course in Numerical Analysis. New York: McGraw-Hill, 1965.

36. Schramm, R., and M. Scott. "Derivative Information Recovery by a Selective Integration Technique." Army Missile Test and Evaluation Directorate Tech Memorandum 1040. White Sands Missile Range: Army Missile Test and Evaluation Directorate, Nov. 1962.

37. Agee, William. "A Multidimensional Fortran Subroutine for DIRSIT." Army Missile Test and Evaluation Directorate Internal Memorandum 3. White Sands Missile Range: Army Missile Test and Evaluation Directorate, Aug. 1965.

38. Foster, T. "Preliminary Documentation of Filter X." Analysis and Computation Directorate Internal Memorandum 40. White Sands Missile Range: Analysis and Computation Directorate, Jan. 1967.

39. Baker, C. "Program Description for ATHENA DCS, Post Flight Programs, DCSFIL and RTARC7." Analysis and Computation Directorate Internal Memorandum 95. White Sands Missile Range: Analysis and Computation Directorate, n.d.

40. McCool, W. A. "A Digital Computer Subroutine for the Automated Selection of Radar Data for Instantaneous Impact Prediction." Computer Directorate Internal Memorandum 8. White Sands Missile Range: Computer Directorate, Feb. 1966.

41. McCool, W. A. "DFX-A New Digital Filter." Analysis and Computation Directorate Internal Memorandum 14. White Sands Missile Range: Analysis and Computation Directorate, Mar. 1966.

42. McCool, W. A. "QD-A New Efficient Digital Filter." Analysis and Computation Directorate Internal Memorandum 60. White Sands Missile Range: Analysis and Computation Directorate, Aug. 1967.

43. McCool, W. A. "QDVSP, An Efficient Digital Data Processing Subroutine." Analysis and Computation Directorate Internal Memorandum 81. White Sands Missile Range: Analysis and Computation Directorate, Feb. 1969.

44. McCool, W. A. "A Matrix Vector Formulation of the QD Filter." Analysis and Computation Directorate Tech Rept 3. White Sands Missile Range: Analysis and Computation Directorate, Dec. 1968. [AD 746623]

45. McCool, W. A. "Zero and First Order QD Filters." Analysis and Computation Directorate Tech Rept 15. White Sands Missile Range: Analysis and Computation Directorate, Mar. 1970.

46. Comstock, D. W., M. Wright, and V. Tipton. "Handbook of Data Reduction Methods." Data Reduction Division Tech Rept. White Sands Missile Range: Data Reduction Division, 13 Aug 1964. [NTIS AD-754857]

47. Blinchikoff, Herman J., and Anatol I. Zverev. Filtering in the Time and Frequency Domains. New York: Wiley, 1976.

48. Papoulis, Athanasios. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1965.

49. Peled, Abraham, and Bede Liu. Digital Signal Processing. New York: Wiley, 1976.

50. Bracewell, Ronald N. The Fourier Transform and Its Applications. New York: McGraw-Hill, 1978.

51. Conover, W. J. Practical Nonparametric Statistics. New York: Wiley, 1980.

52. Weaver, H. Joseph. Applications of Discrete and Continuous Fourier Analysis. New York: Wiley, 1983.

53. Brigham, E. Oran. The Fast Fourier Transform and Its Applications. Englewood Cliffs: Prentice-Hall, 1988.

54. Rabiner, Lawrence R., and Bernard Gold. Theory and Application of Digital Signal Processing. Englewood Cliffs: Prentice-Hall, 1975.

55. Oppenheim, Alan V., and Ronald W. Schafer. Digital Signal Processing. Englewood Cliffs: Prentice-Hall, 1975.

56. Belove, Charles, ed. Handbook of Modern Electronics and Electrical Engineering. New York: Wiley, 1986.

57. Jackson, Leland B. Digital Filters and Signal Processing. Boston: Kluwer Academic Publishers, 1989.

58. Daniels, R. W. Approximation Methods for the Design of Passive, Active, and Digital Filters. New York: McGraw-Hill, 1974.

59. Guillemin, E. A. Synthesis of Passive Networks. New York: Wiley, 1957.

60. Storer, J. E. Passive Network Synthesis. New York: McGraw-Hill, 1957.

61. Gold, Bernard, and Charles M. Rader. Digital Processing of Signals. New York: McGraw-Hill, 1969.

62. Hoel, Paul G. Introduction to Mathematical Statistics. 4th ed. New York: Wiley, 1971.

63. "Comparison of Static and Dynamic Filters." White Sands Missile Range: Analysis and Computation Division, 8 Oct 1970.

64. Rabiner, Lawrence R., and Charles M. Rader. Digital Signal Processing. New York: IEEE P, 1972.

65. Rabiner, Lawrence R., Bernard Gold, and C. A. McGoneal. "An Approach to the Approximation Problem for Nonrecursive Digital Filters." IEEE Trans. Audio Electroacoust. AU-18.2 (1970): 83-106.

66. Herrmann, O. "On the Design of Nonrecursive Digital Filters with Minimum Phase." Electron. Lett. 6.11 (1970): 328-29.

67. Herrmann, O., and H. W. Schuessler. "Design of Nonrecursive Digital Filters with Minimum Phase." Electron. Lett. 6.11 (1970): 329-30.

68. Hofstetter, E., A. V. Oppenheim, and J. Siegel. "A New Technique for the Design of Nonrecursive Digital Filters." Proc. of 5th Annual Princeton Conf. Inform. Sci. Sys. 1971. 64-72.

69. Hofstetter, E., A. V. Oppenheim, and J. Siegel. "On Optimum Nonrecursive Digital Filters." Proc. of 9th Allerton Conf. Circuit Sys. Theory. Oct. 1971.

70. Siegel, J. "Design of Nonrecursive Approximations to Digital Filters with Discontinuous Frequency Responses." Ph.D. thesis, MIT, Jun. 1972.

71. Parks, Thomas W., and James H. McClellan. "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase." IEEE Trans. Circuit Theory. CT-19.2 (1972): 189-94.

72. Parks, Thomas W., and James H. McClellan. "A Program for the Design of Linear Phase Finite Impulse Response Filters." IEEE Trans. Audio Electroacoust. AU-20.3 (1972): 195-99.

73. Rabiner, L. R. "The Design of Finite Impulse Response Digital Filters Using Linear Programming Techniques." Bell Sys. Tech. J. 51.6 (1972): 1177-98.

74. Rabiner, L. R. "Linear Program Design of Finite Impulse Response (FIR) Digital Filters." IEEE Trans. Audio Electroacoust. 20.4 (1972): 280-88.

75. Schwarz, H. R. Numerical Analysis: A Comprehensive Introduction. Chichester: Wiley, 1989.

76. Young, David M. and Robert Todd Gregory. A Survey of Numerical Mathematics. Vol. 1. Reading: Addison-Wesley, 1972.