SECTION 2

DIGITAL FILTER CONCEPTS
&
OVERVIEW OF MANUAL

The primary goal of DISPRO is to help you design a digital filter and evaluate its performance as effectively as possible. This manual will assist you in the reaching of that goal by providing a guide to the features of DISPRO and illustrating some of the ways in which DISPRO can be applied to actual problems. Before you plunge into the details presented in the remainder of this manual, take the time to come along on a short tour of digital filter concepts and the manual contents as presented in the remainder of this section.

2.1 How Much of Your Design Can DISPRO Do?

Selecting a filter type and deciding upon the numerical values that specify its frequency response is pretty much the designer's major task. DISPRO aids by rapidly designing filters, but only after it has been given a set of numerical specifications. For each filter design there is an initial dialogue in which you will be asked for values for the filter bandedge frequencies and the desired ripple/attenuation. DISPRO computes the filter order and allows you to interactively explore the relationships between the various filter parameters, but the final choice is always yours. In making the decision to use a selected filter type, of a particular order, and with a specific set of performance specifications, you must be guided by the requirements of your application. DISPRO makes it easy to try competing designs, so that your final decision can be made on a wide base of information.

Once you have decided upon a filter type and its specifications, all further analysis of the filter properties can be done within DISPRO. This means that all frequency and time behavior can be computed for the filter operating in any wordlength arithmetic, from floating point (24 bit fraction, 8 bit exponent) to your choice of fixed-point/integer (2 to 23 bits+sign). The storage and retrieval of computed data is done automatically by DISPRO, using a DISPRO-created filter data file.

2.2 A Survey of Digital Filter Facts

Digital filters operate on signal samples which are, usually, obtained from an analog signal by means of an analog-to-digital converter. It is the nature of a sampled signal that its spectrum is completely described by the region between d-c and one-half the sampling frequency. Consequently, a digital filter frequency response is specified only for this same region of the frequency axis.

Although digital filters can be described mathematically, this is usually of little use when one is concerned with the larger questions of "What are they?" and "Why use them?". We can start by saying that a digital filter's

[Copyright © Dr. John O'Donnell 2005]

performance is characterized in the same way that an analog filter's performance is: by means of passband and stopband edge frequencies, and by attenuations in the pass and stop bands. Although DISPRO will design FIR digital filters with arbitrary magnitude response shapes, this discussion will be restricted for now to the commonly used lowpass (LP), highpass (HP), bandpass (BP), and bandstop (BS) characteristics.

2.2.1 Why Use Digital Filters?

The major attractions of digital filters, as opposed to analog (including passive, active, and switched-capacitor), are the achievable accuracy and stability, the freedom from tedious tuning and adjustment procedures, and the ability to multiplex several signals through the same filter. Of course, an analog-to-digital converter is required, and the filter operations will probably be done by a microprocessor (or a DSP microprocessor), and a digital-to-analog converter may also be needed—but this seeming complexity can be justified by the achieved precision, stability, and flexibility. Let's consider these benefits in a little more detail:

- *Precision:* the digital filter frequency response accuracy is determined by the arithmetic precision used—floating point certainly gives high precision!—and the accuracy of the system clock. The latter is of importance because a digital filter's critical frequencies are tied to the sampling or update rate.

- *Stability:* the digital filter characteristics can change only if the system clock frequency changes.

- *Flexibility:* many channels of sampled data can be multiplexed through a single filter, and the frequency response of any filter can be changed by calling up and using another set of stored coefficients.

2.2.2 What Kinds Are There?

There are two categories of digital filters: *infinite impulse response (IIR)* or *recursive*, and *finite impulse response (FIR)*, or *nonrecursive*. IIR filters have both poles and zeros, and use past output values—hence "recursive"—as well as the current input value to generate the next output sample. FIR filters have only zeros and generate an output sample from past input samples only—hence "nonrecursive".

Each category has its own design techniques. The IIR filters are actually developed from classical analog filter types, such as Butterworth, Chebyshev, and Elliptic. FIR filters, although strictly practical in digital implementations only, do have a counterpart in the analog world as a tapped delay line with multipliers at the taps. It is usual to consider only that type of FIR filter which has a linear phase, or constant delay, property. The design techniques for linear-phase FIR filters are unique to digital filters.

In the DISPRO software, for example, the user can select from four kinds of IIR filters and two kinds of FIR filters. For the first-time user, and even for the experienced user, the immediate question is which type of filter to use. Not just whether to use IIR or FIR, but also which one of the IIR or FIR types. Let's tackle the first of these questions: IIR or FIR? The answer depends upon whether the phase shift characteristic of the filter is of importance. If phase is not important—and the desire is to have some specified cutoff rate, passband ripple, and stopband attenuation—then the IIR filter must be chosen because of its greater

 [Copyright © Dr. John O'Donnell 2005]

computational efficiency. What this means is that for equal frequency response performance the IIR filter needs fewer arithmetic operations per data sample than does the FIR filter. But there are many applications in data acquisition and communications systems where phase and/or delay distortion must be controlled. In these applications it would be necessary to select the FIR linear phase digital filter, even though it has a greater computational burden.

2.2.3 Finite-Wordlength Considerations

Well, that question seems to be settled now, doesn't it? Not quite—there still remains the matter of the effect of finite precision arithmetic. If the digital filter is to be realized in floating-point arithmetic then there actually is no problem, particularly if the data samples possess 16 or fewer bits of precision. But the most commonly available level of precision is 16 or 24 bits of integer or fixed-point arithmetic. The quantization of digital filter coefficients to the required processor wordlength causes the poles and zeros of the filter to change. Generally, 16-to-24-bit coefficients are sufficiently accurate for most digital filters with reasonably tight performance specifications. In fact, for FIR filters of length less than 50 or 60, experience shows that a coefficient wordlength of 13 bits is usually satisfactory. (This is the wordlength associated with the MPYK instruction on TMS320 processors.)

A different set of problems crops up when IIR filters are implemented in fixed-point arithmetic: scaling of coefficients, and the possibility of computational overflow. It is common, in 2's-complement fixed-point arithmetic, to assume that all numbers are less than unity in magnitude; i.e., the number range is ± 1 and the binary point is assumed (or, in some cases, actually is) between the sign bit and the most significant bit of the value. FIR filter coefficients don't have to be scaled for fixed-point arithmetic—they are always less than unity in magnitude—but IIR filters often have coefficients greater than unity in magnitude. Customarily, IIR filters are realized as a cascade of second-order sections, or biquads. The coefficients of a stable digital filter biquad can be shown to be always less than 2.0 in magnitude. In the DISPRO software there are several different methods provided for the scaling of IIR biquad coefficients. In addition, there are a number of methods for implementing the IIR digital filter equations so as to provide a tradeoff between execution time—as determined by the number of arithmetic operations per data sample—and the sensitivity to overflow. Note that there are some architectures in which the number range is ± 2; for those processors coefficient scaling is not necessary, but the problem of computational overflow must still be considered.

2.2.4 The Types of IIR and FIR Filter Characteristics

And now for the second question: which type of IIR or FIR? The standard types of IIR filter characteristics are Butterworth, Chebyshev types I and II, and Elliptic. The standard FIR designs are the Equiripple and the Kaiser window. Let's discuss the IIR types first.

2.2.4.1 IIR Filter Characteristics

All IIR filters in DISPRO are obtained from normalized analog lowpass prototypes through a band transformation and the bilinear-z mapping.

- *Butterworth:*

  Butterworth filters have maximally flat passbands. They are often used for pulse transmission applications because of the absence of passband ripple (which can cause echoes). In the digital form, with a sampling frequency of FS Hz, there is a high order zero of transmission at FS/2 for lowpass, at d-c for highpass, and at the geometric band center for bandstop. This last property can be used to obtain a notch filter. Suppose the stopband edge frequencies to be specified by FA1 and FA2. The bilinear-z mapping establishes a tangent relationship between critical frequencies in the analog prototype and those in the digital filter; thus it is necessary to consider the "warped" frequencies due to the tangent relationship. Let W1 = tan(π*FA1/FS), and W2 = tan(π*FA2/FS). Define W3 = $\sqrt{}$(W1*W2) as the geometric mean of the warped frequencies. The null will be located at FN = (FS/π)*arctan(W3). For a desired notch frequency FN the values of FA1 and FA2 will have to be determined iteratively.

  Unless the "maximally flat" characteristic—all derivatives of the magnitude response are zero at d-c for a lowpass filter, and at the passband geometric center for a bandpass filter—of the Butterworth is essential it would be more efficient to use a Chebyshev Type II. This type of filter also has a ripple-free passband, and is from 20% to 200% more efficient than the Butterworth. Relative efficiency increases as the transition band decreases in width, equivalently, as the sharpness of cutoff increases.

- *Chebyshev Type I:*

  A Chebyshev filter is more efficient than a Butterworth in the sense that for the same set of specifications the Chebyshev will have a lower order. The Chebyshev Type I filter has an equiripple characteristic in the passband. The passband magnitude fluctuates between 0 dB and -AMAX dB; the number of fluctuations is determined by the filter order. For even orders the response magnitude is -AMAX at zero frequency for lowpass filters, and at FS/2 for highpass filters. The gain at these frequencies is 0 dB for odd order HP and LP filters. A Chebyshev Type I can also be used as a notch filter by designing it as a bandstop filter. The notch frequency is determined in the same way that it was for the Butterworth.

- *Chebyshev Type II:*

  For a given set of specifications the Chebyshev Type I and Chebyshev Type II are of the same order. The Chebyshev Type II, however, has its equiripple characteristic in the stopband. The passband looks similar to that of a Butterworth filter, and for that reason the Chebyshev Type II is often used in pulse transmission applications where it is more efficient than the Butterworth, having lower order for the same performance. Chevyshev type II bandstop filters do not give a notch at the geometric band center. Because the Chebyshev Type II filter has a smooth passband and is of the same order as a Chebyshev Type I filter with identical specifications, it would seem that there is little reason for using a Chebyshev Type I except for notch filtering. In the analog world a Chebyshev Type II, because it has zeros, requires more components in its realization. But in the digital world the single extra multiplication per biquad for the Chebyshev Type II seems a small price to pay for the smooth passband; having ripple in the passband is not normally an *essential* requirement.

 [Copyright © Dr. John O'Donnell 2005]

- *Elliptic:*

  The Elliptic Function filter has the Chebyshev Type I passband and the Chebyshev Type II stopband. It is the most efficient of the IIR filters, and is usually chosen because of the narrow transition band(s) that can be achieved for a modest value of filter order. As is true for the Chebyshev Type II the Elliptic Function filter cannot be used for notch filtering. If the passband ripple is not objectionable then the Elliptic Function filter requires no more computation per sample than does the Chebyshev Type II, and has an efficiency that is from 20% to 60% better than the Chebyshev. Relative to the Butterworth, the Elliptic function filter is from 40% to 380% more efficient.

2.2.4.2 FIR Filter Characteristics

The FIR filters designed by DISPRO have linear-phase characteristics—constant delay. This is achieved by forcing the impulse response of the filter to be symmetric about the central sample. In some hardware and software implementations it may be advantageous to use this symmetry to reduce the number of multiplications by one half. Let's consider the two major types of FIR filter designs.

- *Equiripple:* The most useful type is the Chebyshev Equiripple FIR filter designed using the Parks-McClellan implementation of the Remez exchange algorithm. Before discussing the characteristics of this type of FIR filter it is of value to consider the design method briefly. The procedure is based upon the use of Chebyshev polynomials to approximate an ideal filter characteristic. In the unmodified form this ideal filter frequency response magnitude has a value of 1.0 in each passband, and 0.0 in each stopband (in DISPRO you also have the option to specify any piecewise-linear shape for the magnitude response in any band). To allow a solution to be achieved, it is necessary that the transition bands between the pass and stop bands be of nonzero width. The length of the filter determines the number of Chebyshev polynomial coefficients to be used. Given the desired ideal magnitude characteristic the Parks-McClellan method determines the values of the specified number of Chebyshev polynomial coefficients so that the frequency response of the Equiripple FIR filter exhibits minimum ripple in both the pass and stop bands. The magnitude of this resultant ripple cannot be controlled directly, but is dependent upon the length and band structure of the desired filter.

  In DISPRO the usual LP, HP, BP, and BS filters are specified using a format identical to that for the IIR filters. The length of the filter is computed from an empirical formula based on a large number of design trials. Because the design process is empirical the resulting Equiripple FIR filter characteristic will not match exactly the specifications on passband ripple and stopband attenuation. It usually is necessary to try a few designs in order to satisfy performance requirements. The Parks-McClellan implementation of the Remez exchange algorithm provides a remarkably effective tool for the design of FIR linear phase digital filters. Although the process is iterative in nature, and may occasionally fail, the circumstances of such failures are rare.

- *Kaiser Window:* This type of linear-phase FIR filter is designed by using a Kaiser window function to select a finite number of the coefficients of the Fourier series expansion of the ideal frequency response. Because the frequency response of a digital filter is periodic, with period equal to the sampling frequency, it can be expanded in a Fourier series. The coefficients of this expansion are the samples of

[Copyright © Dr. John O'Donnell 2005]

the impulse response of the digital filter. For ideal LP, HP, BP, and BS filters with zero-width transition bands, the impulse response is a doubly-infinite (from $-\infty$ to $+\infty$) set of time samples with a sin(x)/x characteristic. Simply taking a finite number of these time samples would lead to large amplitude ripple at the transition from pass to stop band, and would give poor stop band performance. By modifying the set of time samples through multiplication by the values of a window, or tapering, function, the ripple magnitude at the pass-to-stop-band transition points can be greatly reduced.

Kaiser has used the properties of his window function to develop a set of design formulas for FIR filters, giving the required length as a function of filter band edges and desired stopband attenuation. The Kaiser window is superior, or at least equal, to all other window functions for this purpose; thus in DISPRO the run-of-the-mill windows, such as Hann, Hamming & Blackman, are not provided because they have nothing to offer.

After some experience designing FIR filters you will discover that the Kaiser window FIR filter is less efficient than the Equiripple FIR filter, in the sense that equal specifications lead to a longer length Kaiser window filter. What then is the reason for using a Kaiser window FIR filter? First, the stopband of the Kaiser-windowed design is not equiripple, but has a -6 dB/octave slope; this may be desirable in some applications, and could actually lead to shorter length designs than are achieved with the Parks-McClellan method if the full stopband attenuation is not required at the beginning of the stopband. Second, when designing very long FIR filters the Kaiser window design computation time is very much shorter than the Equiripple design time, and the Kaiser window design process, not being iterative, always works.

2.2.4.3 Choosing Between IIR and FIR

A final word on the choice between IIR and FIR filters: It may be that there are hardware considerations which affect the choice. This is particularly true because of the number of specialized chips and boards available for FIR filters. However, when a filter is to be implemented on a DSP microprocessor the number of computational steps per unit time may well be the overriding factor—here the IIR filter will usually beat out the FIR filter.

2.3 Module Descriptions and Reference Guide

Section 3 constitutes the heart of the manual, with Section 6 providing a quick reference capability. Each major module within DISPRO is described individually in Section 3 in a format that presents the purpose of the module, its capabilities and parameter ranges, running times, input source(s), outputs, and example of typical use. Perhaps this form may be too detailed for an initial reading, but can best be used as a source of information while running DISPRO.

The short reference guides in Section 6 provide essential information in a compact form for use after some experience with the module has been obtained.

[Copyright ©Dr. John O'Donnell 2005]

2.4 Learning Through Example

Section 4 provides a case study of the filter design and evaluation process. The intention is to illustrate how much you have to bring to the process, and how much DISPRO can aid you. The case study will lead you through the following steps:

- Designing an IIR or FIR filter to meet the application's requirements,
- Computation of frequency responses for floating-point and finite-wordlength coefficients,
- Creating a data file containing an input sample sequence for testing the filter performance,
- Computation of the filter response to the input sample sequence, using finite-wordlength 2's-complement arithmetic.
- Spectral analysis of input and output time sample sequences.
- Comparison of measured and calculated performances.

This case study will provide a broad exposure to the capabilities of DISPRO.

2.5 Customizing Your Use of DISPRO

DISPRO's analysis facilities have been designed to be open for application to any appropriate data set that you create, not just those computed by DISPRO modules. In addition the results provided by DISPRO are available for additional processing to meet your specific needs. The discussion in Section 5 shows you how to interface with DISPRO . All programs listed in Section 5 are also provided on a non-protected diskette that accompanies this manual. Typical uses would be to analyze a filter designed, or modified, by you; plot and spectral analyze time samples created by simulation programs; and use data generated by DISPRO as the input to special processing or simulation software.

2.6 A Word About the Esc Key

In performing filter design and analysis with DISPRO you will be traversing a sequence of menu choices. At almost every point where you are asked to make a decision you are also provided an "escape route." In most cases the "Esc" key will allow you to "escape" the current level of the menu tree and take you back to a higher level. Usually this is explicitly noted on the bottom line of the screen, or in the border of a pop-up menu.

[Copyright ©Dr. John O'Donnell 2005]