LINEAR-PHASE BANDSPLITTING:
THEORY AND APPLICATIONS

2132 (A-1)

James A. Moorer and Mark Berger
Lucasfilm Ltd.
San Rafael, California

Presented at the 76th Convention 1984 October 8-11 New York

AES

This preprint has been reproduced from the author's advance manuscript, without editing, corrections or consideration by the Review Board. The AES takes no responsibility for the contents.

Additional preprints may be obtained by sending request and remittance to the Audio Engineering Society, 60 East 42nd Street, New York, New York 10165 USA.

AN AUDIO ENGINEERING SOCIETY PREPRINT

LINEAR-PHASE BANDSPLITTING: THEORY AND APPLICATIONS

James A. Moorer and Mark Berger

Lucasfilm Ltd.
P.O. Box 2009
San Rafael, Ca 94912

ABSTRACT

There are a number of applications for banks of bandpass filters in professional audio studios, both for film and music production. In this paper, we explore digital techniques for bandsplitting that have the property that the spectrum may be separated into a number of bands such that when these bands are added back together, the result is a pure delay. There need be no amplitude or phase distortion other than delay. This allows such applications as linear-phase graphic equalizers, multi-band noise gates, and many other improvements over conventional studio equipment. These algorithms have been implemented on a large-scale audio signal processor and run in real time. They are currently being used in major motion picture production.

1. Introduction

We will take as the problem to be solved here the case of the graphic equalizer, and the case of *a posteriori* noise reduction. We will show that one manner of solving these problems uses the common mechanism of linear-phase bandsplitting. We will further show that formulating the solutions this way will offer some unique advantages that can only be obtained through digital processing of the audio signal. Finally, we will show the relation between bandsplitting and short-term Fourier analysis with application to FIR bandpass filter design. We begin with a description of the problems to be solved:

1.1. Graphic Equalization

One common complaint with graphic equalizers is that if you set each band to, say, a 6 dB boost, the resulting spectrum is no longer flat. The more the boost or cut, the more "bumpy" the spectrum becomes. Graphic equalizers are generally equivalent to pure gain or attenuation only when all the controls are set at 0 dB boost/cut. Similarly, the various bands of the equalizer interact, so that adjusting one band will often force readjustment of the adjacent bands. We seek a graphic equalizer of potentially any number of bands of any desired widths which has the property that

the bands can be made essentially non-interacting, that the composite transfer response at any setting of the band gains exhibits linear phase, and that when all the band gains are identical, the equalizer is equivalent to pure boost or cut with no amplitude distortion (other than the digital quantization of the signal and possibly some roundoff error).

1.2. Noise Reduction

In modern film production, often dialog recorded on the set must be used, despite the fact that there may be contaminating noise present in the recording. For the most part, this noise consists of wide-band random signals, such as air or wind noise. In many cases, it is not necessary to completely eliminate the noise, but would be sufficient to simply attenuate it between utterances or in quiet passages. The general term for this kind of processing is called "noise gating", where the gain of an amplifier is set to zero (or nearly zero) during the quiet passages. If the noise is not too severe, this is an acceptable method of noise reduction. Furthermore, if the noise is concentrated in a particular spectral range, then it is not necessary to gate the entire signal, but just that portion of the signal in the frequencies where the noise is present. An obvious way to do this is to split the signal into a number of frequency bands and apply noise gating to each one separately. For this application, we require that when the bands are summed, in the absence of noise gating (that is, while the gates are not applying any attenuation), the resulting signal has no distortion other than some small time delay.

We see then that these two applications can make use of the same mechanism: the splitting of the audio spectrum into a number of bands such that they may be recombined by addition to recreate the original signal without error (other than time delay). Let us now see how this may be done.

2. Short-Term Fourier Analysis

One objection to the Fourier transform is that one loses time: that is, when we take the Fourier transform of a signal, the result is a function of frequency only, and not of time. The time variable goes away. To make the Fourier transform correspond more closely to our intuitive (and perceptual) concept of spectrum, we can define a time-variant transform that operates only on a limited part of the signal at each point in time. The result of this is a function of both time and frequency, and can be thought of as a time-varying Fourier transform. More specifically, given a sampled data signal $x(n)$, we define its time-variant short-term Fourier transform, $X(n,m)$, as follows [1]:

$$X(n,m) = \sum_{i=-\infty}^{\infty} h(i-n)x(i)e^{\frac{-2\pi j\, i\, m}{N}} \tag{1}$$

where n is the sample number (time index) and m is the frequency number. For a given value of m, say m_0, the 1-dimensional signal $X(n,m_0)$ represents a function of time only that can be interpreted as the output of one channel of a bank of filters with the center frequency of the band equal to $m\ f_s/N$, where f_s is the sampling frequency in Hertz.

To show that this is a reversible transform, we need only sum over the frequency term, m. Let us do so and call the result $y(k)$.

$$y(k) = \sum_{m=0}^{N-1} X(n,m)e^{\frac{2\pi j\, m\, k}{N}} \tag{2}$$

$$= \sum_{m=0}^{N-1} \sum_{i=-\infty}^{\infty} h(i-n)x(i)e^{\frac{2\pi j\, m\,(k-i)}{N}}$$

By interchanging the order of summation, we derive the following:

$$= \sum_{i=-\infty}^{\infty} h(i-n)x(i) \sum_{m=0}^{N-1} e^{\frac{2\pi j\, m\,(k-i)}{N}}$$

$$= \sum_{i=-\infty}^{\infty} h(i-n)x(i)\, \frac{e^{2\pi j\, m\,(k-i)}-1}{e^{2\pi j\, m\,(k-i)/N}-1}$$

The last term of this equation is zero for all values of $k-i$ except when $k-i$ is an integral multiple of N, where it takes on the value N. This means that $y(k)=x(k)$ if $h(n)$ obeys the following properties:

$$h(0)=\frac{1}{N} \tag{3a}$$

$$h(l\,N)=0, \quad l \neq 0,\ l \ \text{an integer} \tag{3b}$$

This derivation is taken from Portnoff [1].

The point of this is the following: if $h(n)$ represents the impulse response of low-pass filter of any kind that obeys the constraints of equation (3), then $X(n,m)$ is the output of the bandpass filter with a frequency response equal to that of $h(n)$, only shifted up in frequency to $m\ f_s/N$ Hertz. To see how this works, let us take the impulse response of the system. If we take $x(n)$ to be unity at $n=0$ and zero elsewhere, we can see that $X(n,m)$ will be the following:

$$\hat{X}(n,m) = h(-n)e^{\frac{-2\pi j\, n\, m}{N}} \tag{4a}$$

$$= h(-n)\left[\cos\left(\frac{2\pi j\ n\ m}{N}\right) - j\ \sin\left(\frac{2\pi j\ n\ m}{N}\right)\right] \tag{4b}$$

which can be recognized as the impulse response translated in (complex) frequency by an amount $2\pi m/N$.

Note that equation (3) does not constrain the filter $h(n)$ to be high-pass, low-pass, or anything else. It is just that if we want equation (4) to represent a bandpass filter, then $h(n)$ must be a low-pass filter. Additionally, if $h(n)=h(-n)$, the filter will be of linear phase. For the remainder of this paper, we will assume that $h(n)$ is a linear-phase, low-pass filter. In this case, $X(n,m)$ represents the output of N linear-phase, complex bandpass filters. The original signal may be perfectly reconstructed by adding together the output of all these bandpass filters †.

† Note that a linear-phase filter, $h(n)$, is unrealizable as such. If we assume, however, that $h(|n|)=0$ for $n > n_0$, then we can produce a realizable filter simply by replacing $h(n)$ by $h(n+n_0)$, thus introducing an n_0 sample delay.

Note that we will alternately refer to $X(n,m)$ as the output of the m^{th} bandpass filter at sample number n, and as the *transform* of $x(i)$ as a function of the sample number, n, and the frequency index m.

2.1. Whither Linear Phase?

As an aside, we might ask why the linear-phase property is important. In many applications, it makes no difference at all whether the filters are linear phase, zero phase, maximum phase, minimum phase, or whatever. In fact when using a graphic equalizer to counteract the effects of a system that already exhibits phase distortion, linear-phase filters are of no help whatsoever in correcting this phase distortion. On the other hand, if we are using bandsplitting as defined above for a special effect (to "brighten" a passage, for instance), then we want to use linear phase filters to prevent any differential delay between the various channels. The linear phase property assures us that no matter what linear combination of the channels is taken, the result will be linear phase. The high frequencies will emerge at the same time as the low frequencies. This can make a slight difference with impulsive sounds, but the effect in any case would be quite subtle.

2.2. Whither Fourier Analysis?

Although what we described above is certainly a way of splitting a signal into a number of equally-spaced frequency bands such that they may be summed to form a pure delay, one might imagine that there should be other ways to break up the spectrum. In fact, however, if the frequency bands are equally spaced, then *all* other bandsplitting methods can be derived from the time-variant Fourier analysis as shown above.

A more relevant question might be why we insist on equally-spaced bands, where other options, such as constant-Q (third octave), are more common and perhaps more perceptually relevant? The problem is that there is no general closed-form discrete constant-Q transform with the identity property that has been published at this time. There is octave decimation [2], and some other forms, but there is no general analog to equation (1). Note that there is such an analog in the continuous domain [3], and perhaps such a transform will be forthcoming in the future.

Moreover, by combining bands of the above Fourier analysis, we can produce approximations to any grouping we wish. In fact, for octave bands, or any grouping where the width of a band is related to the width of the previous band by a rational number (*i.e.*, a ratio of two integers), then the grouping can be done exactly. We use Fourier analysis as a paradigm for developing this class of bandsplitting filters. When we come to actually realize the filters, as we will show later, we will not have to realize exactly equation (1) above, but we will derive an equivalent form that exhibits much lower compute requirements in certain cases and employs no complex arithmetic.

There are also some reasons related to perception why at least the lower frequencies should be grouped roughly linearly rather than exponentially (constant-Q). This relates to the fact that critical bands, thought to be the the fundamental bandwidth of the ear's frequency analysis, are roughly linearly spaced in the lower frequency range. One might speculate that this is so that harmonics of the speaking voice are separated into distinct bands, but this would be pure conjecture.

3. Choice of Low-Pass Filters

The sharpness of the cutoff of each of the bands depends entirely on the low-pass filter, $h(n)$. The spectral shape of each bandpass filter will be identical to that of the low-pass filter shifted in frequency so that it is centered on the particular band, as shown in equation (4b). Although we could choose any kind of low-pass filter, we will only deal with one particular class of filter which we may define as follows:

$$h(n) = \begin{cases} G \left\{ 1 + \sum_{k=1}^{M} D_k \cos\left(2\pi \frac{n\,k}{N}\right) \right\} & -\frac{N}{2} \le n < \frac{N}{2} \\ 0 & \text{otherwise} \end{cases} \tag{5a}$$

where the gain factor, G, may be set as follows:

$$G = \frac{1}{N} \frac{1}{\left\{ 1 + \sum_{k=1}^{M} D_k \right\}} \tag{5b}$$

This assures that the window satisfies equation (3).

This class includes the popular Hamming [4] ($D_1 = 0.92$), Hanning [4] ($D_1 = 1.0$), and Taylor [5] windows. This exposition of this window family is taken from Rife and Vincent [6]. It does not include, for instance, the Dolph-Chebychev window functions [7] or the Kaiser-Bessel window [8].

The point of chosing this class of window functions is that the N-point discrete Fourier transform of these window functions have *only* $2M+1$ *non-zero points.* In fact, we can find the transform of these windows by use of equation (1). If we again set the input signal, $x(n)$, to the unit impulse, we can obtain the channel-by-channel impulse response as follows:

$$\bar{X}(n,m) = \begin{cases} G \left\{ 1 + \sum_{k=1}^{M} D_k \cos\left(-2\pi \frac{n\,k}{N}\right) \right\} e^{\frac{-2\pi j\, n\, m}{N}} & -\frac{N}{2} \le n < \frac{N}{2} \\ 0 & \text{otherwise} \end{cases} \tag{6a}$$

If we limit n to be between $-N/2$ and $N/2-1$, we may write this as follows:

$$= G \left\{ e^{\frac{-2\pi j\, n\, m}{N}} + \frac{1}{2} \sum_{k=1}^{M} D_k \left[e^{\frac{-2\pi j n(m+k)}{N}} + e^{\frac{-2\pi j n(m-k)}{N}} \right] \right\} \tag{6b}$$

Note that this can be separated into $2M+1$ different components. In fact, it is equivalent to the following 2-step proceedure: first, take a transform, $\underline{X}(n,m)$, using $\underline{a}$ unit window, $h(n)=1$ for $n \ge -N/2$ and $n < N/2$ and $h(n)=0$ elsewhere. We may then form $\bar{X}(n,m)$ as follows:

$$\bar{X}(n,m) = G\left\{ \underline{X}(n,m) + \frac{1}{2} \sum_{k=1}^{M} D_k \left\{ \underline{X}(n,m-k) + \underline{X}(n,m+k) \right\} \right\} \tag{7}$$

This means that for this class of windows, *the output of a particular channel may be realized as a weighted sum of a small number of channels of an analysis which was computed using a simple, rectangular window.* The window need not be applied to the input sequence directly, but may be relized as linear combinations of adjacent transform values. This result is, in fact, obvious, since it is just a restatement of the well-known convolution property: multiplication in the time domain is equivalent to convolution in the frequency domain and vice-versa. What this class of windows accomplishes is to limit the number of frequency terms that need be summed for each channel of

output. With the Kaiser-Bessel window, for instance, the linear combination would include *all* the frequency terms.

It should also be clear that wider bands may be formed by simply summing adjacent bands. Since this class of filters (windows) obeys equation (3), the sum of all bands will always be an identity.

The general procedure, then, for performing linear-phase bandsplitting is first to use equation (1) with a rectangular unit window function to produce N analysis channels, then use (7) above to combine the channels to produce the desired bandwidth and sideband rejection. The problem is then reduced to finding the coefficients, D_k, that produce the desired band shapes.

As is usual in engineering, there are no "best" choices for windows: there are only tradeoffs of various kinds. Fortunately, this problem is equivalent to one that has been thoroughly studied, and that is the problem of the design of directional antennas [5, 7]. We will suggest here one class of windows that represents a "reasonable" compromise between the desire for sharp cutoff and high stop-band rejection. These are called Class III weighting functions by Rife and Vincent [6], and are shown in Table I.

	k			
M	1	2	3	4
1	1.0			
2	1.19685	.19685		
3	1.43596	.497537	.0615762	
4	1.566272	.725448	.180645	.0179211

TABLE I - Window coefficients, D_k of the Class III weighting functions [6] for various orders, M.

The window for $M = 1$ is, of course, the Hanning window.

Note that equation (5a) as it is written does not exactly represent a linear-phase filter. If N is even, then the cosines must be shifted by 1/2 sample to make the window function perfectly symmetric. If N is odd, then the limits must be changed to $-(N-1)/2 \le n \le (N-1)/2$. This assures linear phase. We have been somewhat careless in the use of $N/2$ above, since it is ambiguous when N is odd.

3.1. The Graphic Equalizer Revisited

We can see now that by simply multiplying each channel of our analysed signal, $X(n,m)$, by a constant, say $a(m)$, we get the effect of a graphic equalizer. The property of linear phase follows automatically if $h(n)$ is linear phase. The flatness is automatic if the $a(m)$ are equal. Furthermore, if (3) is obeyed, the rolloff (sharpness) of the bandpass filters may be made arbitrarily great by different choices of $h(n)$.

3.2. The Noise Gate Revisited

We may implement various kinds of *a posteriori* noise reduction by first dividing the signal into a number of frequency bands, then by applying a dynamic range expansion algorithm to the output of each band. The idea of the expander is to reduce the gain of an individual band when the energy in the band drops below a certain level. The exact manner in which the gain is reduced (*i.e.*, how abrupt the transition is) determines the properties of the expander. If a very sharp transition from unity gain to zero gain is introduced, the result is a noise gate. Many other choices are possible. Our choice is to use the following expansion function [9]:

$$g = (1-\alpha)\left(\frac{s}{s+\lambda}\right)^{2r} + \alpha \qquad (8)$$

s is the RMS signal coming out of a particular band. λ is the threshold where the gain will start to be reduced. r determines the sharpness of the gain change. If r is set to a large value (like 5), a noise gate is realized. A more reasonable value is $r = 1$. α determines the minimum gain at low levels. With α set to, say, -12 dB, the signal during the quiet portions will be attenuated by 12 dB. With α set to one, no expansion is performed.

Since computing s necessarily involves some delay, we delay the channel output by that same amount before applying the gain control. This gives the channel the effect of seeing into the future, since the gain starts changing well before the signal arrives. Delays are trivially implemented in digital form. Note that this makes the more common "fast attack, slow decay" adjustments much less important, since they are simply a way of making up for the fact that the signal is not known in advance. Using this kind of "look-ahead" makes the gain change, say, at the beginnings of spoken words, especially plosives, much more subtle. The common "bite" of the noise gate opening up is greatly reduced.

In actual usage, one has to decide how many bands to use and what their response should be. At first, one might think that the more independent frequency bands, the better the results will be. This is only partially true. It turns out that there must be either very few bands (such as 4 bands), or very many (256 or more) and nothing in between seems to help. The problem seems to be that with intermediate numbers of bands, each band covers just one or two critical bands. When the gain in a particular band is reduced, the "missing" spectrum is quite audible. It sounds a bit like trans-oceanic radio broadcasts fading in and out. With a small number of bands, the coverage is overlapping so that each critical band in the ear receives contributions from several different filter bands. This means that when one band drops out, there is still some contribution from the other bands, so that the "phasing" effect is reduced. With a very large number of bands, each critical band in the ear receives contributions from several different filter bands, thus making the loss of a single filter band less audible. With intermediate values, the size of the filter bands and the size of the critical bands become of roughly the same order, producing highly audible results. The preceeding is a plausibility argument only and should not be mistaken for established fact.

The best results we have obtained are with large numbers of bands. We break the signal into 256 bands and apply the gain control to each one separately. The threshold, λ, is set during a "training" run on just the noise alone. This gives us the RMS value of the noise in each band, σ_m. This will be recognized as an approximation to the standard deviation of the noise in that band. We then set the threshold to some multiple of the standard deviation. Setting the threshold to 5 times σ_m eliminates all noise, but will also eliminate the more quiet vocal sounds, such as terminal sibilants. A threshold of $2.5\sigma_m$ gives a reasonable tradeoff between noise reduction and quality. $1.5\sigma_m$ was used in the movie "Amadeus" to preserve the maximum performance nuance in the dialog, giving only a slight but noticable reduction in the noise.

We should mention that the 256-band system does produce a curious artifact. It replaces the noise with a swishing, semi-musical tonality. It is a consequence of the random nature of noise that occaisional excursions above the threshold are probable from time to time. Since these excursions are isolated and limited to a very narrow frequency band, the result is a distinct pitched sound. With this many bands, there are numerous "clouds" of these pitched sounds. In general, we find it necessary to cover these artifacts with some other masking sound, either by adding a

bit of the original noise back in, or making sure that there is music or sound effects present to mask the artifacts.

In the 256-band noise reducer, no combining of channels was done. The expansion algorithm was applied to each and every channel. For the 4-band reducer, the spectrum was first broken into 16 equally-spaced bands, giving a frequency resolution of 1548 Hz ($N = 31$, 48000 Hz sampling rate). A window was not used in this case since a great deal of overlap was desired. The upper two bands involved the merging of a large number of bands. Table II shows the exact channel gains that were used for each of the four bands.

CHANNEL	FREQUENCY	GAIN			
		1	2	3	4
0	0	.55	.27	.11	.07
1	1500	.31	.38	.19	.12
2	3000	.25	.33	.25	.17
3	4500	.17	.26	.35	.22
4	6000	.10	.18	.43	.29
5	7500	.05	.15	.45	.35
6	9000		.11	.45	.44
7	10500		.07	.42	.51
8	12000			.40	.60
9	13500			.37	.63
10	15000			.35	.65
11	16500			.34	.66
12	18000			.33	.67
13	19500			.33	.67
...					

TABLE II - Channel coefficients for the 16 channels of each of the bands in the 4-band noise reducer.

Note that in each channel, the sum of the gains is always unity. This assures that with equal band gains, the frequency response of the system is a constant (plus a delay). That is, there is no "coloration" introduced by the system itself.

There are other reasons for chosing the amount of overlap shown in Table II. The problem is that often the "crispness" of speech seems to be contained in low-amplitude, high-frequency signals. These signals can be of such low amplitude that they do not cause the gain of the expander to rise, and consequently are lost. By including crossfeed from the low channels, we are assured that while speech is present, even the high bands will be open, and consequently all the high transients will be preserved. The high-end overlap between band 3 and band 4 shown in Table II is deliberate, so that when one or the other of the gains goes to zero, there will still be some feedthrough. This prevents too abrupt a transition. We note in passing that the filters in the Dolby A noise reduction system are similar to this, although the low band is at a much lower frequency in the Dolby unit. By adjusting these channel gains, one can adjust exactly the tradeoff between preserving the high frequency, low-amplitude information, and reducing the noise. With only four bands, this is about as good as can be done. With 256 or more bands, we can do better, since we can actually reduce the background noise that is *between* individual partials of the signal.

This kind of noise reduction is not new [9, 10, 11, 12], but our implementation (which will be discussed next) is unique.

4. Implementation

Although we could just compute equation (1) directly, there are a number of ways to reduce the computation while still preserving the desirable properties of the function. Portnoff [1] showed that the fast Fourier transform may be used to compute (1), which reduces the operations considerably. Furthermore, he showed that since the channel outputs are band-limited, they need not be computed at every sample, but only every R samples. The limit of this is to use a Hamming or Hanning window and set $R = 1/2$. This provides the minimum of computation using the method of Portnoff. By use of polyphase filtering [12], this computation may be reduced even further, at the expense of introducing a slight (arbitrarily small) ripple in the frequency response of the system.

The only problem with this kind of implementation is that when the gain of a band is changing rapidly, there can be a "modulation" introduced in the band that is not simply the gain changing. It is the sum of the window functions, separated by R samples, multiplied by differing values of gain, and summed. That is, there is an additional distortion due to the fact that we are computing the transform every R samples *while* the gain is changing. Ideally, we should compute the channel outputs *every* sample so that time-variant operations, like gain changes, can be made as smooth and free of artifacts as possible. Our experimental results hint that for best results R can be no greater than 2 without introducing artifacts.

With this in mind, we shall look at alternative realizations that provide this property.

4.1. FIR Filter Implementation

One can perform the analysis directly by filtering the input signals with a set of bandpass filters with impulse responses given by equation (4). Recall also that we may sum these channels any way we please without changing the key features of the system - *i.e.*, that the system is an identity and that it is linear phase. One convenient form is to define a new, all-real channel output $y(n,m)$ as follows:

$$y(n,m) = X(n,m) + X(n,N-m) \qquad 0 < m < N/2 \tag{9}$$

This yields the following impulse response for each channel:

$$y(n,m) = \begin{cases} h(-n) & m = 0 \\ 2\,h(-n)\cos\left(\frac{2\pi j n m}{N}\right) & 0 < m < N/2 \\ (-1)^n\, h(-n) & m = N/2, \quad N \quad even \end{cases} \tag{10}$$

Consequently, the filters for the 4-band noise reducer above may be calculated directly from the gains given in Table II. Since these filters are perfectly symmetric, only $N/2$ multiplications and N additions are required for each band. In the 4-band noise reducer example, $N = 31$. The filter coefficients for each of the four bands are given in Table III. The output of each band may then be calculated as follows:

$$y(n) = \sum_{p=-15}^{p=15} a_p\, x(n-p) \tag{11}$$

Note that this involves future values of $x(n)$. To make this filter fully realizable, we delay $x(n)$ by 15 samples, giving the following formula:

$$y(n) = \sum_{p=-15}^{p=15} a_p\, x(n-p-15) \tag{12}$$

This is now fully realizable. Note that in Table III, summing the four filters together gives just a single impulse (ignoring roundoff error in the coefficients). That is, the sum of a_0 is one and the sum of the others is effectively zero. This shows that the response of the sum of the filters is exactly flat. Figure 1 shows the responses of the individual filters. The dotted line above shows the response of the sum of the filters (note that the decibel scale makes it harder to see the summation property clearly).

Note what has happened here. We have used the model of time-variant Fourier analysis to develop a model of linear-phase bandsplitting. We have actually realized the model through direct convolution. In this case, the Fourier transform is not used in the implementation. For a small number of bands, it is almost always more efficient to use direct convolution, which has the further advantage that the output of each band is produced at the original sampling rate (*i.e.*, $R = 1$). For larger numbers of bands, the efficiency gained by the use of the fast Fourier transform cannot be ignored.

COEFF	BAND			
	1	2	3	4
a_0	0.077000	0.107667	0.358333	0.490333
a_1	0.069551	0.085069	-0.009671	-0.144950
a_2	0.050938	0.038429	-0.042735	-0.046632
a_3	0.029869	0.005568	-0.035372	-0.000065
a_4	0.014456	-0.000931	-0.014834	0.001309
a_5	0.007727	0.001994	-0.005205	-0.004516
a_6	0.007230	-0.001415	-0.001192	-0.004624
a_7	0.008338	-0.007619	0.000098	-0.000817
a_8	0.008116	-0.007674	-0.000076	-0.000366
a_9	0.006735	-0.003638	-0.000799	-0.002298
a_{10}	0.005931	-0.002737	-0.001671	-0.001523
a_{11}	0.006596	-0.004661	-0.001464	-0.000472
a_{12}	0.007873	-0.004381	-0.002328	-0.001164
a_{13}	0.008305	-0.002475	-0.003372	-0.002458
a_{14}	0.007503	-0.003433	-0.002307	-0.001764
a_{15}	0.006497	-0.006430	-0.001404	0.001338

Table III - FIR filter coefficients for each of the bands of the 4-band noise reducer. The coefficients a_{-1} through a_{-15} are just the same as a_1 through a_{15}, *i.e.*, the filters are symmetric.

4.2. A Remark on Filter Design Theory

We note here that this gives us a design methodology for FIR filters where the object is to separate the spectrum into distinct bands such that the sum of those bands is unity. These bands may be of any general shape. For instance, we may produce two filters, one of which is a bandpass filter and the other of which is a bandstop filter, such that together they include the whole spectrum. By use of the weighting functions, we may make the cutoff rate arbitrarily sharp (at the expense, of course, of lengthening the filter). To see how this is done, let us first consider just placing a window function on each analysis channel. This involves weighting the channel outputs with the D_k as shown in equation (7). We may then group the weighted outputs any way we please to produce a myriad of combinations of band separating filters. We show this schmatically in Table IV.

BAND	CHANNEL					
	m	$m+1$	$m+2$	$m+3$	$m+4$	$m+5$
k	D_2					
$k+1$	D_1	D_2				
$k+2$	1	D_1	D_2			
$k+3$	D_1	1	D_1	D_2		
$k+4$	D_2	D_1	1	D_1	D_2	
$k+5$		D_2	D_1	1	D_1	D_2
$k+6$			D_2	D_1	1	D_1
$k+7$				D_2	D_1	1
$k+8$					D_2	D_1
$k+9$						D_2

Table IV - Weights for different analysis bands to increase sharpness of the bands. Any group of bands may now be summed to produce wider bands with the same cutoff.

Any combination of rows in Table IV may be summed, such as from $k+2$ to $k+4$, to produce a filter with wider passband. If we sum the rows from, say, k_0 to k_1, we can express the result in closed form as follows:

$$y(n,m) = \frac{\sin\left[m\frac{\theta}{2}(k_1 - k_0 + 1)\right]}{\sin(m\frac{\theta}{2})} \cos\left[m\,\theta\frac{1}{2}(k_0 + k_1)\right] \left\{1 + \sum_{k=1}^{M} D_k \cos(k\,\theta)\right\} \quad (13)$$

where $\theta = 2\pi n/N$. The terms of this formula can be interpreted as the ideal low-pass filter, shifted in frequency by heterodyning so that it is centered on the desired band (*i.e.*, at $1/2\,(k_0 + k_1)$), and multiplied by the window function itself. This is simply the closed form expression for the time-limited impulse response produced by the windowing method [13]. The only revelation here is that by summing other groups of bands, an ensemble of filters can be produced that split up the spectrum and sum to unity.

4.3. Frequency Sampling Realization

For intermediate numbers of channels, another option is viable, and this is the frequency sampling filter [4]. To derive this, we start with the band impulse response we desire, $s(n)$. We can then define the Z-transform of the impulse response as follows:

$$S(z^{-1}) = \sum_{n=0}^{N-1} s(n)\, z^{-n} \quad (14)$$

We may also define the discrete Fourier transform of $s(n)$ and its inverse as follows:

$$S(m) = \sum_{n=0}^{N-1} s(n) e^{-j\,2\pi\frac{n\,m}{N}} \quad (15a)$$

$$s(n) = \frac{1}{N}\sum_{m=0}^{N-1} S(n) e^{j\,2\pi\frac{n\,m}{N}} \quad (15b)$$

If we substitute (15b) into (14), after some manipulation, we get the following:

$$S(z^{-1}) = \frac{1}{N}(1 - z^{-N}) \sum_{m=0}^{N-1} \frac{S(m)}{1 - z^{-1} e^{2\pi j\frac{m}{N}}} \quad (16)$$

This derivation is taken from Rabiner and Gold [13]. If we then combine terms with index k with terms with index $N - k$ (except for $k = 0$ and, when N is even, except for $k = N/2$), assuming that $S(k) = S(N-k)$, which is a direct consequence of the linear phase property, we get the following:

$$S(z^{-1}) = \frac{1}{N}(1 - z^{-N}) \left[S(0)\frac{1}{1 - z^{-1}} + S(N/2)\frac{1}{1 + z^{-1}} + 2\sum_{m=1}^{\frac{N}{2}-1} S(m) \frac{1 - \cos(2\pi\frac{m}{N})z^{-1}}{1 - 2\cos(2\pi\frac{m}{N})z^{-1} + z^{-2}} \right], \quad N \text{ even} \quad (17a)$$

$$S(z^{-1}) = \frac{1}{N}(1 - z^{-N}) \left[S(0)\frac{1}{1 - z^{-1}} + 2\sum_{m=1}^{\frac{N-1}{2}} S(m) \frac{1 - \cos(2\pi\frac{m}{N})z^{-1}}{1 - 2\cos(2\pi\frac{m}{N})z^{-1} + z^{-2}} \right], \quad N \text{ odd} \quad (17b)$$

It is hard to convey the importance of this formulation. The significance is that *any* FIR filter may be realized as the sum of a number of resonators fed from a comb filter. This means that the output of each of these resonators may be identified with $X(n,m)$ as defined in equation (1)! The impulse response of any one of the resonators is a pure cosine of N samples duration, which will be recognized as the signal in equation (10)†.

† Note that unlike in equation (1), for this derivation, there is a hidden assumption that $h(n)$ is limited in time to N samples or less. That is, that $h(n)$ is zero for $n < 0$ and for $n \geq N$. The condition in equation (3) is then trivially satisfied.

To aid numerical stability, it is convenient to replace z^{-1} by $r\ z^{-1}$ for $r < 1$ to bring the poles of the resonators within the unit circle. Values of r from 0.9 to 0.9999 have been used successfuly, but must be carefully chosen depending on the word length of the arithmetic being used and the filter length N.

Once the bands are split up in this manner, they may be recombined using the Class III weighting functions as described above to provide quite steep cutoff filters. With this implementation, it is still true that when we sum all the bands together, the response is identically unity, except for a delay. There is no frequency distortion at all. This is the implementation we are using for our graphic equalizer and it seems to be perfectly well accepted for studio use for medium and high frequencies. It is not as well adapted for lower frequencies (less than, say, 400 Hertz) because the number of resonators required becomes prohibitive. It is more convenient to use the fast Fourier transform in this case.

Note that the coupled form of the digital resonator [14] is ideal for this case. Figure 2 shows the flow diagram of the filter, where $C = \cos(2\pi m/N)$ and $S = \sin(2\pi m/N)$. If we let $w(n)$ represent a state variable of the filter, initialized to zero at the beginning of time, then we may write the recursion equations for this filter as follows:

$$y(n) = x(n) + r\cos(2\pi\frac{m}{N})y(n-1) - r\sin(2\pi\frac{m}{N})w(n-1) \tag{18a}$$

$$w(n) = r\sin(2\pi\frac{m}{N})y(n-1) + r\cos(2\pi\frac{m}{N})w(n-1) \tag{18b}$$

This form automatically realizes the zero of transmission and has superior low-frequency roundoff properties, both for the coefficients and for noise amplification of the signal itself.

Since we are necessarily limited in space in this exposition, we will not resolve the conflict between the definition of $h(n)$ in equation (5a), which was centered on $n = 0$, and the fact $s(n)$ in equation (14) is summed only from $n = 0$ to $n = N - 1$. Since this is just a phase change, we leave it as an exercise to the interested reader to follow the details through.

5. Summary and Conclusion

In this paper, we have used short-term time-variant Fourier analysis as a vehicle to develop a family of methods of separating the spectrum into a number of bands such that the sum of these bands is exactly unity plus a constant delay. We have mentioned methods of realizing this process using the fast Fourier transform, but have also derived here additional realizations using either FIR filtering or IIR filtering by frequency sampling filters, both utilizing purely real arithmetic. Any of these realizations can be used where bandsplitting is required.

Two applications have been mentioned which use this theory as a basis. These were graphic equalization and noise reduction. For graphic equalization, all that is required is to multiply the output of each band by the desired gain before summing the bands. For noise reduction, a dynamic-range expander is placed after each band filter which attenuates the signal at low levels. At high levels, the signal is passed unchanged. We noted that for best results, 256 or more bands should be used so that noise between the partials of the signal may be attenuated. A specific example of a 4-band noise reducer was given that may be used like conventional noise gates or other expanders.

All the techniques described above have now been used in major motion picture production (though not without some difficulty at times due to the experimental nature of the research) and are now available for wider application in the audio industry in general.

6. References

[1] M. Portnoff, "Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. ASSP-24, pp243-248 (1976 June)

[2] R. Crochiere and L. Rabiner, *Multirate Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, New Jersey, 1983)

[3] G. Gambardella, "A Contribution to the Theory of Short-Time Spectral Analysis with Nonuniform Bandwidth Filters," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp455-460 (1971, July)

[4] B. Gold and C. Rader, *Digital Processing of Signals* (McGraw-Hill, New York, 1969)

[5] T. Taylor, "Design of Line-Source Antennas for Narrow Beamwidth and Low Side Lobes," *IRE Trans.*, Vol AP-3, pp16-28 (1955 January)

[6] D. Rife and G. Vincent "Use of the Discrete Fourier Transform in the Measurement of Frequencies and Levels of Tones," *The Bell System Technical Journal*, February 1970

[7] C. Dolph, "A Current Distribution for Broadside Arrays which Optimizes the Relationship between Beam Width and Side-Lobe Level," *Proc. IRE*, vol 34, pp335-348 (1946, June)

[8] J. Kaiser, "Digital Filters," in *System Analysis by Digital Computer* (John Wiley & Sons, Inc., New York, 1966) pp218-285

[9] T. Eger, J. Su and L. Varner "A Nonlinear Spectrum Processing Technique for Speech Enhancement," *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, Vol 2, San Diego, paper 18A.1 (1984, March)

[10] J. Borish, "Electronic Simulation of Auditorium Acoustics," PhD Thesis, Stanford University (1984) Dept. of Electrical Engineering

[11] J. Strawn, *Personal Communication* (1977-present)

[12] R. Lagadec and D. Pelloni, "Signal Enhancement via Digital Signal Processing," 74th Audio Eng. Soc. Convention, preprint 2037 (G-6) (New York, 1983, October)

[13] L. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, New Jersey, 1975)

[14] A. Oppenheim and C. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proceedings of the IEEE*, Vol 60, pp 957-976 (1972, August)

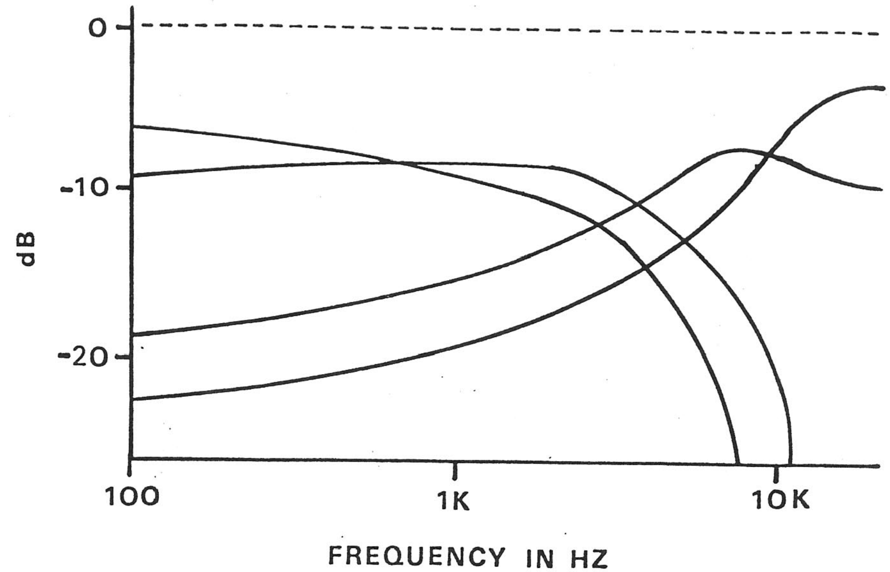

Figure 1 - Frequency responses of the four filters in the 4-band noise reducer. The dotted line above shows the (linear algebraic) sum of the responses of the four filters. It is, of course, perfectly flat. Note the deliberate overlap of the bands, especially in the high shelving region.

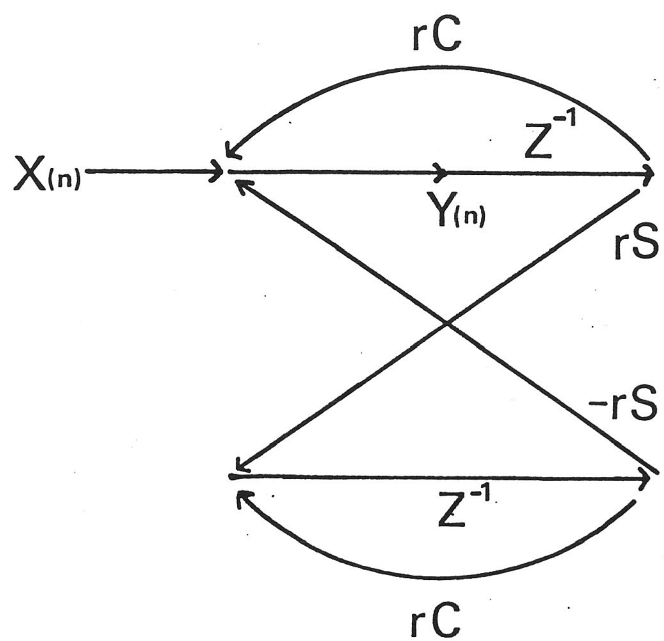

Figure 2 - Signal flow graph of the coupled form of the digital resonator. This is a very numericaly stable form of the second-order digital filter with the additional property that a zero of transmission is created that is exactly what is needed to form frequency-sampling filters.