Design of Hybrid Filter Banks for Analog/Digital Conversion

Scott R. Velazquez, Truong Q. Nguyen, *Senior Member, IEEE,* and Steven R. Broadstone

***Abstract*— This paper presents design algorithms for hybrid filter banks (HFB's) for high-speed, high-resolution conversion between analog and digital signals. The HFB is an unconventional class of filter bank that employs both analog and digital filters. When used in conjunction with an array of slower speed converters, the HFB improves the speed and resolution of the conversion compared with the standard time-interleaved array conversion technique. The analog and digital filters in the HFB must be designed so that they adequately isolate the channels and do not introduce reconstruction errors that limit the resolution of the system. To design continuous-time analog filters for HFB's, a discrete-time-to-continuous-time ("Z-to-S") transform is developed to convert a perfect reconstruction (PR) discrete-time filter bank into a near-PR HFB; a computationally efficient algorithm based on the fast Fourier transform (FFT) is developed to design the digital filters for HFB's. A two-channel HFB is designed with sixth-order continuous-time analog filters and length 64 FIR digital filters that yield -86 dB average *aliasing* error. To design discrete-time analog filters (e.g., switched-capacitors or charge-coupled devices) for HFB's, a lossless factorization of a PR discrete-time filter bank is used so that reconstruction error is not affected by filter coefficient quantization. A gain normalization technique is developed to maximize the dynamic range in the finite-precision implementation. A four-channel HFB is designed with 9-bit (integer) filter coefficients. With internal precision limited to the equivalent of 15 bits, the maximum *aliasing* error is -70 dB, and with the equivalent of 20 bits internal precision, maximum *aliasing* is -100 dB. The 9-bit filter coefficients degrade the stopband attenuation (compared with unquantized coefficients) by less than 3 dB.**

I. Introduction

FILTER banks are used in a number of communications applications such as subband coders for speech signals [1], frequency-domain speech scramblers [2], and image coding [3]. Fig. 1 illustrates an M-channel maximally decimated parallel filter bank, where H_k and $F_k, 0 \leq k \leq M-1$ are analysis and synthesis filters, respectively. The analysis filters H_k channel the input signal $u[n]$ into M subband signals, which are downsampled by a factor M. The downsampler, which decreases the sampling rate of the signals, and the upsampler, which increases the sampling rate of the signals, are denoted by the boxes with down and up arrows, respectively, as shown in Fig. 1. The synthesis filters F_k reconstruct the signal. The theory for perfect-reconstruction (PR) nonhybrid filter banks has recently been established [4]–[7].

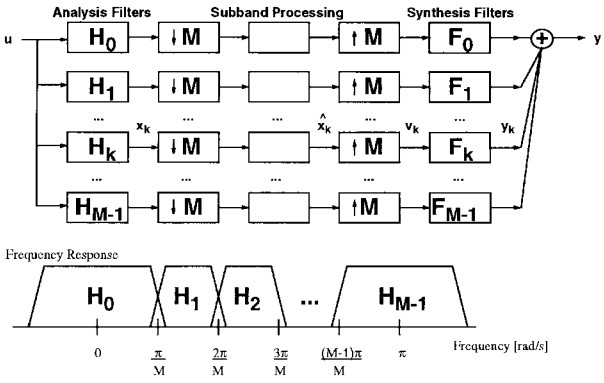

Fig. 1. M-channel, maximally-decimated, parallel, filter bank.

Using a filter bank for analog-to-digital (A/D) or digital-to-analog (D/A) conversion is an unconventional application of the filter bank architecture that improves the speed and resolution of the conversion over the standard time-interleaved array conversion technique. The hybrid filter bank analog-to-digital converter (HFB ADC) uses analog analysis filters to allocate a frequency band to each ADC in the array and digital synthesis filters to reconstruct the digitzed signal. Similarly, the reverse structure, which is known as the hybrid filter bank digital-to-analog converter (HFB DAC), employs digital analysis filters and analog synthesis filters. The HFB significantly improves the speed and resolution of the conversion by attenuating the effects of mismatches between the converters in the array, which otherwise severely limit the resolution of the system.

For very high-speed conversion, manufacturers such as Hewlett-Packard and Tektronix have conventionally implemented time-interleaved ADC's that consist of an array of M moderate speed ADC's, which are triggered successively at $1/M$ the effective sample rate of the system. The speed and resolution of the system is limited because the system is extremely sensitive to ADC mismatches and clock timing errors. With signal preconditioning and compensation for linearity and timing errors, Hewlett-Packard has been able to build an 8 GSa/s, 8-bit, time-interleaved ADC with a signal bandwidth of nearly 2 GHz [9]. Effects of mismatches in converters in time-interleaved systems has been studied extensively [10], [11].

Petraglia and Mitra applied a fully discrete-time quadrature mirror filter (QMF) bank to A/D conversion by using switched-capacitor analysis filters, downsamplers, an array of ADC's

Manuscript received February 15, 1997; revised November 30, 1997. This material is based on work supported under a National Science Foundation Graduation Fellowship. The associate editor coordinating the review of this paper and approving it for publication was Dr. Keshab Parhi.

S. R. Velazquez is with The V Company, Revere Beach, MA 02151 USA.
T. Q. Nguyen is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA.
S. R Broadstone is with Teratech Corporation, Burlington, MA 01803 USA.

Publisher Item Identifier S 1053-587X(98)02607-5.

sampling at the subband rate, upsamplers, and digital synthesis filters, as shown in Fig. 1 (where each "subband processing" block is an ADC) [12]. The use of switched capacitors limits the speed of the system typically to the hundreds of kiloherz. In addition, switched-capacitor filters introduce switching noise, which can limit the signal-to-noise ratio (SNR) of the system. Typical switched capacitors have an SNR of about 85 dB [13].

The HFB architectures presented here overcome the time-interleaved architecture's extreme sensitivity to converter mismatches and timing errors and the switched-capacitor architecture's speed and noise limitations. The filters in the HFB architecture isolate the converters in the array and attenuate the *aliasing* errors caused by gain and phase mismatches. A comparison of the effects of gain, phase, and dc-offset errors in the time-interleaved and hybrid filter bank architectures is presented in [8].

Applications of high-speed, high-resolution A/D and D/A conversion include direct digital receivers for wireless communications applications; radar receivers; test equipment such as oscilloscopes, spectrum analyzers, network analyzers, signal and pattern generators; modems; and medical imaging systems.

The practical challenge in constructing an HFB is to compensate for the effects of hardware components whose characteristics (e.g., resistance or capacitance) are imprecise and drift with time and temperature M, which is a challenge not faced with conventional fully digital filter banks. An automated calibration algorithm that compensates for analog filter and converter mismatches and drift is developed in [8].

The first challenge in constructing an HFB is to design the analog and digital filters in the filter bank to provide adequate channel separation and accurate reconstruction of the converted signal. As with conventional filter banks, the HFB itself can introduce gain and phase *distortion* and *aliasing* error into the system, even if the subband conversion is distortionless. Proper design of the filters will minimize the errors introduced by the HFB while still providing good filter characteristics (i.e., sharp cutoff and large stopband attenuation). Efficient design algorithms of both the digital and analog filters are developed in this paper. Section II provides a description of HFB's and the benefits of this architecture. Section III highlights the difference between the design and the calibration of HFB's. Sections IV and V present algorithms for the design of HFB's as well as design examples and results. Section VI summarizes the results.

II. Hybrid Filter Banks

The maximally decimated HFB ADC employs analog analysis filters H_k to channel the wideband analog input signal into M subband signals. Note that the analysis filters shown in Fig. 1 can be continuous-time filters $H_k(s)$ (e.g., passive LC circuits or active operational amplifier circuits), which is called a continuous-time HFB [8], [14], or the analysis filters can be discrete-time filters $H_k(z)$ (e.g., CCD devices, switched-capacitors), which is called a discrete-time HFB. The subband signals are sampled at $1/M$ the effective sample rate of the system and converted to digital signals with n-bit ADC's. The digitized subband signals are upsampled by M and reconstructed via the digital synthesis filters $F_k(z)$. The effective sample rate of the system is M times that of the subband ADC's in the array, and the resolution is n bits, which is the same as that of the subband ADC's in the array.

The discrete-time HFB ADC employs discrete-time analog analysis filters such as CCD devices and discrete-time digital synthesis filters. This discrete-time CCD architecture is well suited for low-power, monolithic, relatively high-speed conversion (tens of megahertz), and moderate resolution (8–12 bits) (e.g., 40 MSa/s, 12-bit HFB ADC chip in CCD technology) [15]. Applications of the discrete-time HFB include medical imaging, portable electronics, and HDTV.

The continuous-time HFB ADC employs continuous-time analog analysis filters such as RLC networks and discrete-time digital synthesis filters. The continuous-time system is well suited for very high-speed (hundreds of megahertz), very high-resolution (8–16 bits) systems that are not necessarily low power (e.g., a 160 MSa/s, 12-bit HFB ADC system using four Analog Devices AD9042 12-bit 40 MSa/s ADC's).

The focus of this paper is the maximally decimated HFB, but it is worth noting that the oversampled HFB ADC can be used to improve the resolution at the expense of the speed of the conversion. The analog analysis filters H_k channel the wideband analog input signal into M subband signals. The subband signals are sampled at L/M times the effective sample rate of the system and converted to digital signals with n-bit ADC's, where L is the oversampling factor, and $L \geq 1$. For each doubling of L, the quantization noise is reduced by 3 dB, or one half of an effective bit. Rate changers alter the digitized subband signals by a factor of M/L, and the output signal is reconstructed via the digital synthesis filters $F_k(z)$. The effective sample rate of the system is M/L times that of the subband ADC's in the array and has a resolution equal to $n + \frac{1}{2}\log_2 L$ effective bits [16].

The HFB DAC is simply the reverse of the HFB ADC and employs digital analysis filters $H_k(z)$ and analog synthesis filters (either continuous-time $F_k(s)$ or discrete-time $F_k(z)$).

The analysis and synthesis filters in the HFB attenuate the effects of gain and phase mismatches between the converters in the array, which otherwise severely limit the resolution of the conventional, time-interleaved architecture. In addition, the HFB requires a single timing signal as opposed to the very accurately time-skewed signals necessary in time interleaving. The analysis and/or synthesis filters in the HFB can be used to compensate for gain and phase errors. Continuous-time filters or CCD devices can be used to overcome the speed and noise limitations of a switched-capacitor implementation.

For specific applications, higher resolution ADC's can be assigned to frequency bands where higher dynamic range is necessary. The HFB requires more hardware than time interleaving, but for applications such as compression or adaptive-array processing, the HFB can directly provide channelization for subband processing without the need for additional filtering hardware. Nonuniform subband bandwidths can be used. The HFB is amenable to a full VLSI implementation to decrease the size, improve the performance, and reduce the cost, especially in a low-power CCD implementation. Polyphase structures can be used to filter the signals at the lowest possible rate.

Since this architecture does not mix high-frequency bands down to baseband, the HFB (like the time-interleaved system) does not overcome the need for high-speed, high-precision sample-and-hold circuitry. Recently, manufacturers such as Analog Devices and Motorola have introduced off-the-shelf converters whose sample-and-hold analog bandwidth far exceeds the converter's Nyquist bandwidth, eliminating the need to implement additional sample-and-hold circuitry for use in the HFB [17].

III. Design Versus Calibration

The design and the calibration of HFB's are considered to be two separate problems. The design procedure calculates the analysis and synthesis filters that provide adequate signal channelization (i.e., high stopband attenuation and sharp cutoff) with accurate signal reconstruction (i.e., negligible *distortion* and *aliasing* error introduced by the filter bank). The design procedure does not take into account the mismatches between the ADC's in the array, and it assumes that the analog filters can be built to specification.

The calibration procedure alters filters to compensate for ADC mismatches and frequency response deviations in the analog filters due to hardware components whose characteristics (e.g., resistance or capacitance) are imprecise and drift with time and temperature variation. An efficient calibration procedure is developed in [8].

IV. Design of Continuous-Time Hybrid Filter Banks

This section presents the design of near-PR continuous-time analog analysis filters and the digital synthesis filters in the maximally decimated continuous-time HFB ADC. Ideally, the analysis filters $H_k(s)$ and synthesis filters $F_k(z)$ are designed such that the HFB output $Y(e^{j\omega})$ should simply be a scaled, delayed version of the input $U(j\Omega)$

$$\begin{aligned} Y(e^{j\omega}) &= \sum_{m=0}^{M-1} U(j(\omega M - 2\pi m)/T) T_m(e^{j\omega}) \\ &= M e^{-j\omega d} U(j\Omega) \end{aligned} \tag{1}$$

where d is the system delay, T is the subband ADC sample period, and the *distortion/aliasing* functions $T_m(e^{j\omega})$ are

$$\begin{aligned} T_m(e^{j\omega}) &= \sum_{k=0}^{M-1} F_k(e^{j\omega}) H_k(j(\omega M - 2\pi m)/T) \\ &= \begin{cases} M e^{-j\omega d} & m = 0 \\ 0 & m = 1, 2, \cdots, M-1. \end{cases} \end{aligned} \tag{2}$$

$T_0(e^{j\omega})$ is the *distortion* function and corresponds to the gain and phase of the filter bank, and $T_m(e^{j\omega}), 1 \le m \le M-1$, are the *aliasing* functions and correspond to the *aliasing* errors in the filter bank. Equation (2) corresponds to the traditional PR conditions. The goal in the design of CT HFB's is to design filters that have the required channelization properties and approximate the PR conditions as closely as possible. *Distortion* should be small (e.g., less than a tenth of a decibel deviation from ideal 0 dB), and *aliasing* error should be minimized so that it does not limit the resolution of the system.

A. Design of Analysis Filters

The first step in designing the analysis filters is to determine the filter specifications necessary to yield the desired speed and resolution of the analog-to-digital conversion (e.g., 160 MSa/s speed with 12-bit resolution). The ADC's in the array are selected to have the desired resolution of the overall system at the highest speed available; based on the speed of the converters, the number of channels M in the HFB is determined (e.g., four 40 MSa/s converters are needed to achieve a system speed of 160 MSa/s). Then, the uniform bandwidth filter cut-off frequencies (-3 dB frequencies) are $(1/2T_sM), (2/2T_sM), \cdots ((M-1)/2T_sM)$ [in Hertz], where T_s [in seconds] is the effective sample period of the system.

Since the HFB attenuates the effects of mismatches between ADC's in the array, the filter *stopband* attenuation is related to the ADC mismatch errors. Mismatches cause *aliasing* errors that limit the *spurious free dynamic range (SFDR)*. The relationship is derived in [8], and the result for gain mismatches in the ADC's is

$$\text{SFDR} = \frac{1}{E\{a^2\} 2(\text{stopband})^2} \tag{3}$$

where $E\{a^2\}$ is the expected value of the gain mismatch error between ADC's in the array. For example, assume a good 12-bit ADC has SFDR $= 80$ dB [18] and that the ADC's have mismatch error $E\{a^2\} = -60$ dB. Therefore, from (3), the required filter *stopband* attenuation is -27 dB. The filter cutoff should be as sharp as possible to make the transition bands as narrow as possible, which simplifies the calibration of the HFB.

One option for the analysis filters is to simply use off-the-shelf filters that meet the required specifications. This is the simplest solution, but as discussed in [19] and [20], the characteristics of some standard filter types complicate the reconstruction demands of the synthesis filters. For example, Chebyshev or elliptic filters can have passband ripple and nonconstant group delay, which require synthesis filters to be very high order to provide accurate signal reconstruction, and Butterworth filters, which have no passband ripple and nearly constant group delay; these enable lower order synthesis filters to provide accurate signal reconstruction.

An alternative is to constrain the number of poles and zeros in the analysis filters and iteratively optimize the pole and zero locations to provide both adequate channelization and accurate signal reconstruction with low-order synthesis filters. This method is discussed in [19].

1) Z-To-S Transform: Another option for the continuous-time analysis filters is to transform a PR discrete-time solution that meets the filter specifications into the continuous-time domain. Relatively low-order discrete-time filter banks have been discovered that introduce no *distortion* or *aliasing* errors and whose filters meet the requirements derived above. The transform should yield stable continuous-time filters that match the frequency response of the discrete-time filter point by point. To preserve the reconstruction accuracy in the design of HFB's, accurate frequency response matching is critical. Neither the bilinear transform nor impulse invariance [21]

techniques meet this criteria; therefore, a new transform similar to the Padé approximation [22] is presented here.

A discrete-time to continuous-time (i.e., "Z-to-S") transform $z^{-1} = G(s)$ converts a discrete-time filter $H(z)$ into a continuous-time filter $\hat{H}(s)$, whose frequency response $\hat{H}(j\Omega)$ accurately approximates that of the discrete-time filter $H(e^{j\omega})$. The transform is a ratio of polynomials in s

$$z^{-1} = G(s) = \frac{G_B(s)}{G_A(s)}. \tag{4}$$

The accuracy of the approximation can be improved by increasing the order of the numerator and denominator polynomials $G_B(s)$ and $G_A(s)$.

To design the analysis filters, a PR discrete-time filter bank is first designed using standard PR discrete-time filter bank design techniques to meet the desired specifications derived above [23]. Second, the transform is used to calculate the continuous-time analysis filters

$$\hat{H}_k(s) = H_k(z)|_{z^{-1}=G(s)} \tag{5}$$

that approximate their discrete-time counterparts.

2) Design of Z-to-S Transforms: The objective is to design the transform $G(s)$, which can be used to directly calculate a continuous-time filter $\hat{H}(s)$, whose frequency response accurately approximates that of a given discrete-time filter $H(z)$. That is, find $G(s)$ such that the error

$$\varepsilon_H = \int_{-\pi}^{\pi} |\hat{H}(j\omega T) - H(e^{j\omega})|^2 \, d\omega \tag{6}$$

is minimized, where T is the effective sample period of the system. $\hat{H}(s)$ and $H(z)$ are related as in (5). $\hat{H}(s)$ should be stable if $H(z)$ is stable.

With the given error criteria, the optimal $G(s)$ is a ratio of polynomials in seconds whose frequency response $G(j\Omega)$ approximates $z^{-1} = e^{-j\omega}$. For simplicity (and with no loss of generality), assume the sampling period $T = 1$ (in seconds) so that $\Omega = \omega T = \omega$; the continuous-time frequency Ω is equal to the discrete-time frequency ω. The design of $G(s)$ is the same as designing a continuous-time filter whose frequency response $G(j\omega)$ approximates $z^{-1} = e^{-j\omega}$ on the interval $-\pi \leq \omega < \pi$ or $0 \leq \omega < \pi$ if the coefficients are real. Algorithms for this standard filter design problem are common (e.g., Matlab function *invfreqs*). The optimization routine calculates the real-coefficient numerator and denominator polynomials of $G(s)$ that minimize the error

$$\varepsilon_G(\omega) = |G(j\omega) - e^{-j\omega}|^2 \tag{7}$$

at N (e.g., $N = 1024$) equally spaced frequencies on $0 \leq \omega < \pi$. The order of the numerator and denominator $G_B(s)$ and $G_A(s)$ can be specified independently. Higher order transforms improve the accuracy of the approximation but increase the order of the resulting filter. For example, a fourth-order transform $G(s)$ [i.e., the highest order of $G_B(s)$ and $G_A(s)$ is 4] transforms an nth order discrete-time filter into a continuous-time filter of order $4n$.

TABLE I
ACCURACY OF ALLPASS Z-TO-S TRANSFORMS

Order of $G(s)$	Average Error [dB]	Standard Deviation of Error [dB]	Maximum Error [dB]
1 (Bilinear Transform)	-8.2	-9.5	-4.4
2	-36.2	-34.6	-25.8
3	-70.9	-68.9	-59.0
4	-110.4	-111.4	-101.8
5	-152.5	-152.2	-140.3

3) Allpass Z-to-S Transforms: Constraining $G(s)$ to be a causal, stable allpass function assures that transformed filters will be stable. Considering allpass functions is natural since the frequency response of the desired transform is $e^{-j\omega}$, which has unity gain. The allpass constraint for $G(s)$ with real coefficients is

$$G_B(s) = G_A(-s) \tag{8}$$

so that the magnitude of $G(s)$ is forced to be unity. The design of allpass transforms requires fewer optimization variables since the numerator and denominator of $G(s)$ are directly related [see (8)] as opposed to an unconstrained optimization, which allows the numerator and denominator functions to be chosen independently (which is considered below).

Table I shows the average, standard deviation, and maximum of the error in five causal, stable, allpass Z-to-S transforms, where the error is defined as in (7).

If $G(s)$ is a causal, stable allpass function (i.e., its poles are in the left half plane), then stable discrete-time filters (i.e., their poles are inside the unit circle) are transformed into stable continuous-time filters. A stable discrete-time pole at

$$z = z_0, |z_0| < 1 \tag{9}$$

is transformed into continuous-time poles at $s = s_0$ via the transform [see (4)] so that $z_0^{-1} = G(s_0)$.

From (9)

$$|G(s_0)| > 1. \tag{10}$$

From the properties of allpass functions, if $|G(s_0)| > 1$, then $\Re\{s_0\} < 0$; the transformed pole locations are in the left half plane, and the transformed filter is therefore stable.

4) Unconstrained Z-to-S Transforms: Allpass functions are constrained to have the same order numerator and denominator [see (8)], which may result in unnecessarily complex transformed filters. The unconstrained transform allows independent specification of the numerator and denominator order. For stability, the order of the numerator should be less than or equal to that of the denominator; otherwise, the magnitude of the frequency response $|G(j\omega)|$ approaches infinity as the frequency increases. Table II shows the average, standard deviation, and maximum value of the

TABLE II
ACCURACY OF UNCONSTRAINED Z-TO-S TRANSFORMS

Order of $G_B(s)$	Order of $G_A(s)$	Average Error [dB]	Standard Deviation of Error [dB]	Maximum Error [dB]
1	3	-32.9	-33.1	-24.3
1	4	-46.6	-46.1	-36.4
2	6	-103.5	-102.4	-91.4
3	5	-109.2	-108.3	-97.3
3	6	-128.8	-127.8	-116.4
4	5	-132.2	-131.2	-119.9
4	6	-151.8	-151.1	-139.3
3	7	-148.5	-147.4	-135.7
3	8	-142.8	-141.5	-129.8

error in nine stable Z-to-S transforms, where the error is defined as in (7).

The unconstrained optimization converges to the allpass solution when the order of the numerator and the denominator are the same, indicating that the allpass function is the optimal solution given the squared-error criteria [see (7)].

Note that not all combinations of numerator and denominator order lead to stable transforms (i.e., some stable discrete-time filters can be transformed into unstable continuous-time filters) unless the transform is allpass. For discrete-time filters $H(z)$ that are causal and have a finite impulse response (FIR), the poles of $G(s)$ should be in the left half plane so that the transformed filter $\hat{H}(s)$ [as defined in (5)] has poles in the left half plane and, therefore, is stable. For discrete-time filters that have an infinite impulse response (IIR), determining if the poles of transformed continuous-time filter $\hat{H}(s)$ will lie in the left half plane is most easily done with numerical Monte Carlo techniques.

A Monte Carlo analysis was performed for all combinations of numerator and denominator order in $G(s)$ up to order ten to determine which transforms yield stable filters. A given $G(s)$ is used to transform a discrete-time filter with a pole at $z = z_0, |z_0| < 1$ into a continuous-time filter, and the poles of the resulting filter are calculated to determine if they are in the left half plane and, therefore, stable. The process is repeated over a large number of random, uniformly distributed pole locations z_0 to provide a good approximation of the locations and percentage of unstable cases.

The Monte Carlo analysis confirms that the order of the numerator $G_B(s)$ should be less than the order of the denominator $G_A(s)$ for the transform to yield stable filters. For all the cases listed in Table II, all poles of $G(s)$ are in the left half plane so that FIR filters are transformed into stable continuous-time filters. Results from the Monte Carlo analysis reveal that a very high percentage of the resulting continuous-time pole locations using the cases listed in Table II were stable; therefore, the vast majority of stable IIR filters are transformed into stable continuous-time filters. The unstable filters were cases when poles near the unit circle in the z domain were transformed into poles just to the right of the $j\omega$ axis in the s domain. When starting with IIR discrete-time filters, one should always confirm the stability of filters designed with this Z-to-S transform.

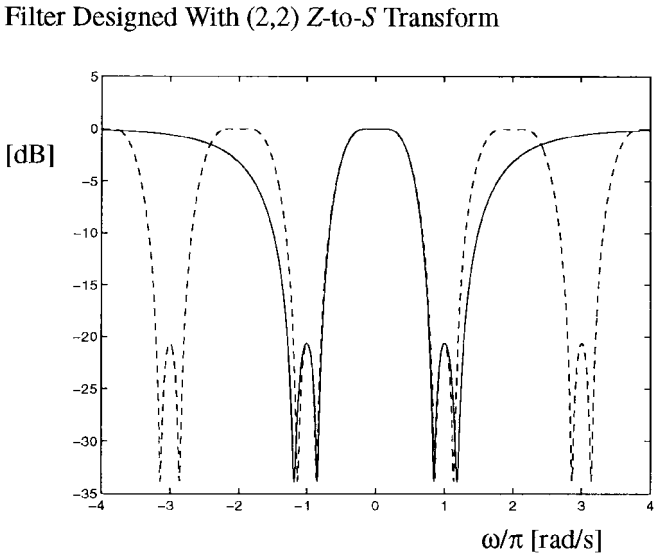

Fig. 2. Sixth-order continuous-time filter (solid) designed with (2,2) Z-to-S transform compared with original third-order discrete-time filter (dashed). Note that frequency response matching is necessary only on $-\pi \leq \omega < \pi$ as shown.

Good transforms for designing HFB's should be stable, low order, and fairly accurate (better than approximately -25 dB maximum error). The following cases meet these criteria [where (B, A) refers to the order of the numerator $G_B(s)$ and denominator $G_A(s)$]: (2,2), (1,3), (1,4), (4,4). The (2,2) transform is the most practical since it is the lowest order (therefore, lowest hardware complexity), and it provides a maximum error of -26 dB, which is about as accurate as a hardware implementation could be built (i.e., even if the transform had better accuracy, hardware component variations would make it difficult to build filters whose frequency response matches the specified frequency response closer than about -25 dB absolute error). The synthesis filters designed in Section IV-B compensate for the inaccuracy of the transform, and the calibration of the HFB compensates for inaccuracy in a hardware implementation. Recall that a kth-order transform translates an nth-order discrete-time filter into a continuous-time filter with order nk so that one drawback of the (2,2) transform is that it yields continuous-time filters with double the order of the original discrete-time filter.

5) Design Results: The allpass (2,2) Z-to-S transform is

$$G(s) = \frac{s^2 - 5.0s + 10.3}{s^2 + 5.0s + 10.3}. \tag{11}$$

As depicted in Table I, this transform approximates $z^{-1} = e^{-j\omega}$ with an average error of -36 dB and maximum error of -26 dB. Note that the approximation is only valid for $-\pi \leq \omega < \pi$, which is the only frequency region needed by the HFB. Fig. 2 depicts the accuracy of the approximation for $-4\pi \leq \omega < 4\pi$.

Unlike the bilinear transform, which maps discrete-time frequencies to continuous-time frequencies with a nonlinear function, this Z-to-S transform attempts a linear mapping,

thereby preserving the frequency response. For the second-order transform above, $\omega = 0$ rad/s in the discrete-time domain translates to $\Omega = 0$ Hz in the continuous-time domain, and $\omega = \pi$ rad/s translates to $\Omega = \sqrt{10.3} \approx \pi$ Hz.

Using well-known discrete-time filter bank design techniques, a two-channel, third-order discrete-time PR FIR filter bank was designed, yielding two third-order discrete-time analysis filters $H_0(z)$ and $H_1(z)$ and two third-order discrete-time synthesis filters $F_0(z)$ and $F_1(z)$. The Z-to-S transform was used to transform the discrete-time analysis filters into two sixth-order continuous-time analysis filters $\hat{H}_0(s)$ and $\hat{H}_1(s)$ via (5). This HFB (which was comprised of sixth-order continuous-time analysis filters and third-order FIR synthesis filters) yields 2.83×10^{-2} dB average deviation from 0 dB *distortion* and -43 dB average *aliasing*. The optimal synthesis filters designed in Section IV-B significantly improve the *distortion* and *aliasing* error.

Using the same third-order prototype PR FIR filter bank above, a new HFB was designed with the allpass (4,4) Z-to-S transform

$$G(s) = \frac{s^4 - 18.8s^3 + 166s^2 - 771s + 1543}{s^4 + 18.8s^3 + 166s^2 + 771s + 1543}. \tag{12}$$

The resulting HFB (12-order continuous-time analysis filters and third-order FIR synthesis filters) yields 1.5×10^{-5} dB average deviation from 0 dB *distortion* and -119 dB average *aliasing*.

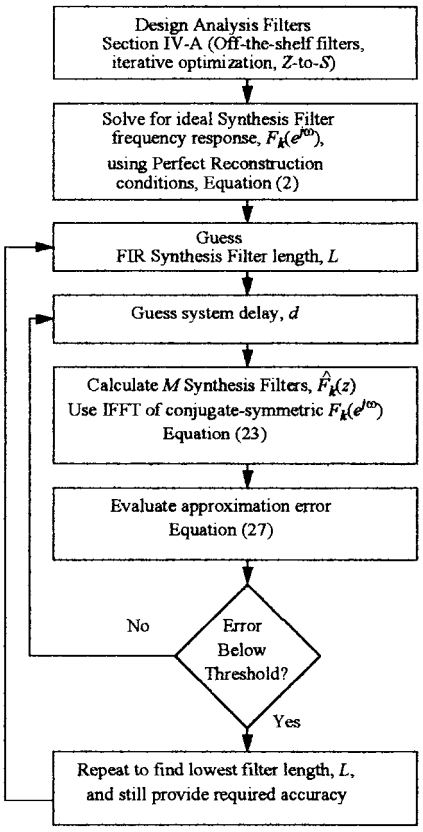

Fig. 3. Synthesis filter design algorithm based on inverse FFT.

B. Design of Synthesis Filters

Given the analysis filters (which are designed with any of the methods described above: standard off-the-shelf analog filters, constrained iterative optimization, or Z-to-S transform), the synthesis filters are designed to minimize the reconstruction error as measured by the *distortion* and *aliasing* functions (2). The *distortion* should be small (e.g., less than a tenth of a decibel deviation from ideal 0 dB), and the *aliasing* should be less than the desired SFDR of the system (e.g., 80 dB for a good 12-bit ADC). This section presents a computationally efficient algorithm that iteratively adjusts the system delay d [as defined in (1)] and calculates the optimal FIR synthesis filters with a squared-error criteria. Fig. 3 shows the flowchart for this synthesis filter design algorithm.

1) Design of Synthesis Filters Using Inverse Fast Fourier Transform: Given the system delay d and the analysis filters $H_k(s)$, the PR constraints in (2) can be solved for the frequency response of the ideal synthesis filters $F_k(e^{j\omega})$. Equation (2) forms a set of M simultaneous equations linear in the M unknown synthesis filters, which can easily be solved with standard linear algebra techniques. For example, the frequency responses of the ideal synthesis filters for $M = 2$ are in (13) and (14), shown at the bottom of the page, on the interval $0 \leq \omega < \pi$, where T is the sample period of the subband ADC's.

This section presents a method for designing the length L FIR synthesis filters $\hat{F}_k(z) = \Sigma_{n=0}^{L-1} \hat{f}_k[n] z^{-n}$, whose frequency responses $\hat{F}_k(e^{j\omega})$ optimally approximate the ideal PR frequency responses $F_k(e^{j\omega})$ in a squared-error sense

$$\varepsilon_k = \int_{-\pi}^{\pi} |\hat{F}_k(e^{j\omega}) - F_k(e^{j\omega})|^2 \, d\omega. \tag{15}$$

To find the minimum, this error function is differentiated with respect to the impulse response $\hat{f}_k[n]$. Using the general relation $|a|^2 = aa^*$, (15) can be expanded to

$$\varepsilon_k = \int_{-\pi}^{\pi} \hat{F}_k(e^{j\omega})\hat{F}_k^*(e^{j\omega}) - \hat{F}_k(e^{j\omega})F_k^*(e^{j\omega}) - F_k(e^{j\omega})\hat{F}_k^*(e^{j\omega}) + F_k(e^{j\omega})F_k^*(e^{j\omega}) \, d\omega. \tag{16}$$

Using the definition of the Fourier transform

$$\frac{d}{d\hat{f}_k[n_0]} \hat{F}_k(e^{j\omega}) = \frac{d}{d\hat{f}_k[n_0]} \sum_{n=-\infty}^{\infty} \hat{f}_k[n] e^{-j\omega n} = e^{-j\omega n_0}. \tag{17}$$

$$F_0(e^{j\omega}) = e^{-j\omega d} T \frac{H_1(j2(\omega - \pi)/T)}{H_0(j2\omega/T)H_1(j2(\omega - \pi)/T) - H_1(j2\omega/T)H_0(j2(\omega - \pi)/T)} \tag{13}$$

$$F_1(e^{j\omega}) = e^{-j\omega d} T \frac{H_0(j2(\omega - \pi)/T)}{H_1(j2\omega/T)H_0(j2(\omega - \pi)/T) - H_0(j2\omega/T)H_1(j2(\omega - \pi)/T)}. \tag{14}$$

Differentiating the error yields

$$\frac{d\varepsilon_k}{df_k[n_0]} = \int_{-\pi}^{\pi} \hat{F}_k^*(e^{j\omega})e^{-j\omega n_0} + \hat{F}_k(e^{j\omega})e^{j\omega n_0} - F_k^*(e^{j\omega})e^{-j\omega n_0} - F_k(e^{j\omega})e^{-j\omega n_0}\, d\omega = 0. \tag{18}$$

The definition of the inverse Fourier transform is

$$\hat{f}_k[n_0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{F}_k(e^{j\omega})e^{j\omega n_0}\, d\omega. \tag{19}$$

Assuming real-valued impulse responses [i.e., $\hat{f}_k[n] = \hat{f}_k^*[n]$ and $F_k^*(ej\omega) = F_k(e^{-j\omega})$], (18) can be simplified to

$$4\pi\hat{f}_k[n_0] - 2\int_{-\pi}^{\pi} F_k(e^{j\omega})e^{j\omega n_0}\, d\omega = 0. \tag{20}$$

Therefore, the impulse response of the synthesis filter that minimizes the squared error is simply the inverse Fourier transform of the ideal frequency response

$$\hat{f}_k[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F_k(e^{j\omega})e^{j\omega n}\, d\omega \tag{21}$$

which can be calculated with the computationally efficient fast Fourier transform (FFT) algorithm.

When calculating the inverse Fourier transform of the ideal frequency response expressions, care must be taken to ensure that the synthesis filters calculated have real coefficients. The Fourier transform must be conjugate-symmetric for its impulse response to be real. The Fourier transform of the ideal synthesis filters are specified over the interval $0 \le \omega < \pi$ as in (13) and (14); therefore, the conjugate-symmetric Fourier transform of the ideal synthesis filters from $-\pi \le \omega < \pi$ (or equivalently $0 \le \omega < 2\pi$ as used in some FFT algorithms) can be formed by taking the complex conjugate of the Fourier transform on $0 \le \omega < \pi$, reversing it in frequency, and moving it to $-\pi \le \omega < 0$ (or $\pi \le \omega < 2\pi$). Furthermore, the symmetry and 2π periodicity properties of conjugate-symmetric Fourier transforms require the imaginary part of the transform to be zero at frequencies that are integer multiples of π, this should be forced manually since the ideal synthesis filter expressions [e.g., (13) and (14)] are not guaranteed to meet this property.

Therefore, (21) [where the frequency response of the ideal filters $F_k(e^{j\omega})$ is forced to be conjugate-symmetric] yields the real-coefficient impulse response of the synthesis filters that minimizes the squared error (15). Equation (21) can be evaluated numerically with the computationally efficient FFT algorithm. Using N samples of the ideal conjugate-symmetric synthesis filter Fourier transform $F_k(e^{j\omega})|_{\omega=(2\pi p'/N)}, p = 0, 1, \cdots, N-1$, the N-point inverse FFT $f_k^{(N)}[n]$ is the impulse response of the desired filter time-aliased every N points. N should be chosen large enough that the impulse response has sufficiently decayed so that the time *aliasing* is negligible (e.g., $N = 1024$ points); therefore

$$f_k^{(N)}[n] \approx \frac{1}{2\pi}\int_{-\pi}^{\pi} F_k(e^{j\omega})e^{j\omega n}\, d\omega = f_k[n]. \tag{22}$$

However, the resulting N-point impulse response is unnecessarily long; therefore, it is windowed with a length L boxcar function

$$\boxed{\hat{f}_k[n] = f_k^{(N)}[n]\, w[n]} \tag{23}$$

where the boxcar function is

$$w[n] = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & L \le n \le N \end{cases}. \tag{24}$$

Equation (23) represents a general, computationally efficient filter design algorithm that calculates the optimal, length L, real-coefficient FIR filter $\hat{f}_k[n]$, whose frequency response matches a desired frequency response $F_k(e^{j\omega}) = A_{\text{desired}}(e^{j\omega})$ with a squared-error criterion.

2) Iterative Optimization of System Delay: The design of HFB's requires specification of the system delay d [as defined in (1)], which can be optimized to further reduce the squared error. The system delay d must be the same for all M synthesis filters; therefore, the joint error function should be used to optimize the delay for all the synthesis filters simultaneously

$$\varepsilon = \sum_{k=0}^{M-1} \varepsilon_k = \sum_{k=0}^{M-1} \int_{-\pi}^{\pi} |\hat{F}_k(e^{j\omega}) - F_k(e^{j\omega})|^2\, d\omega. \tag{25}$$

Evaluation of the error is easier in the time domain than in the frequency domain. By Parseval's relation, minimizing (25) is equivalent to minimizing

$$\varepsilon = \sum_{k=0}^{M-1} \sum_{n=-\infty}^{\infty} (\hat{f}_k[n] - f_k[n])^2. \tag{26}$$

From (22) and (23)

$$\boxed{\varepsilon = \sum_{k=0}^{M-1} \sum_{n=L}^{N} (f_k^{(N)}[n])^2.} \tag{27}$$

The error is minimized by iteratively adjusting the system delay d, recalculating the synthesis filters [see (23)], and repeating until the energy in the truncated coefficients [see (27)] is minimized. Standard minimization algorithms, such as Matlab's $fmin$, can be used. Note that the system delay is not necessarily an integer. This optimization is computationally efficient because it relies on iterated evaluations of the inverse FFT, and it can calculate optimal synthesis filters and system delay in a matter of seconds.

The synthesis filter length L should be the smallest value that still yields the desired *distortion* and *aliasing* error. As mentioned earlier, the *distortion* should be small (e.g., less than a tenth of a decibel deviation from ideal 0 dB), and the *aliasing* should be lower than the desired SFDR of the system. In practice, the filter length L should be chosen such that the *aliasing* should be 6 to 9 dB lower than the desired SFDR to allow for the increase in *aliasing* that occurs when implementing the synthesis filters with quantized coefficients.

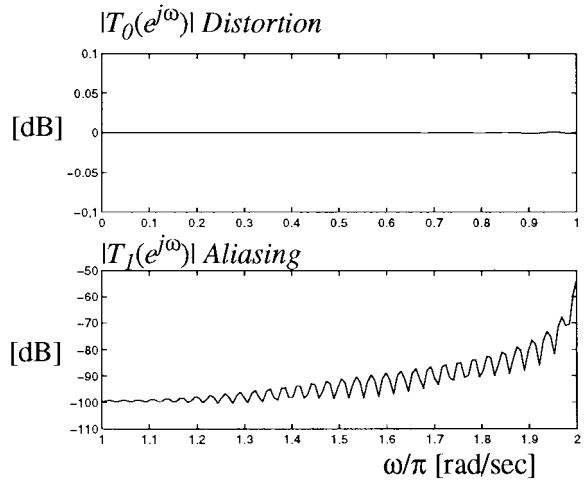

Fig. 4. *Distortion* and *aliasing* functions for CT HFB with sixth-order analysis filters designed with Z-to-S transform and length 64 FIR synthesis filters designed with algorithm in Section IV-B.

3) Synthesis Filter Design Results: Using the sixth-order analysis filters designed with the (2,2) Z-to-S transform in Section IV-A, the synthesis filter design algorithm was used to calculate the optimal system delay d, and the optimal length $L = 64$ FIR synthesis filters. The optimal system delay was $d = 49.9675$. This HFB has 2.7×10^{-4} dB average deviation from 0 dB *distortion* and -86 dB average *aliasing*, as shown in Fig. 4. This HFB would be suitable for systems up to approximately 12 bits resolution since the *aliasing* error is below the SFDR of a good 12-bit device.

The 12th-order analysis filters designed with the (4,4) Z-to-S transform in Section IV-A were accurate enough that further optimization of the original PR prototype synthesis filters is probably unnecessary. As stated in Section IV-A, this CT HFB with 12th-order analysis filters and length $L = 4$ FIR synthesis filters has 1.5×10^{-5} dB average deviation from 0 dB *distortion* and -119 dB average *aliasing*. This system would be would be suitable for systems up to approximately 16 bits resolution.

V. Design of Discrete-Time Hybrid Filter Banks

Due to the analog implementation, the design of discrete-time HFB's faces several obstacles not found in traditional discrete-time filter banks. Discrete-time analog filters such as switched-capacitors or CCD's have multiplier coefficient quantization constraints, limited internal precision, and clipping or overflow that are not limiting issues in a traditional digital signal processor (DSP) implementation of discrete-time filter banks.

This section presents the design of DT HFB's with quantized coefficients that provide near-PR even with limited internal precision. The technique of lossless factorization of paraunitary filter banks in [24] and [25] is modified to allow for limited multiplier coefficient precision, limited internal precision, and clipping. An existing paraunitary PR discrete-time filter bank (such as a cosine-modulated PR filter bank [23]) is factored into lossless factors, each of which can be quantized to any precision without altering the PR property.

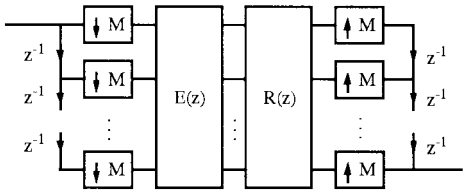

Fig. 5. Discrete-time filter bank shown in matrix form with polyphase filtering performed at subband rate.

The result is a PR filter bank (i.e., no *distortion* or *aliasing*), where the filter frequency responses are approximations of the original unquantized filters with degraded stopband attenuation and passband ripple, depending on the precision of the quantization.

The resulting filter bank is implemented in hardware with finite-precision arithmetic, which increases the *distortion* and *aliasing* error, and is therefore not considered a PR filter bank. Finite-precision arithmetic causes arithmetic operations to be rounded to the internal wordlength of the hardware: an error that accumulates from stage to stage in the system. However, the error can be minimized to an acceptable level by gain normalization between stages to maximize the dynamic range without overloading the signal.

A four-channel example architecture based on a length-16 FIR PR cosine-modulated filter bank is investigated.

A. Lossless Factorization

The analysis filters are represented in matrix form as

$$\begin{bmatrix} H_0(z) \\ \cdots \\ H_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{00}(z^M) & \cdots & E_{0,M-1}(z^M) \\ \cdots & \cdots & \cdots \\ E_{M-1,0}(z^M) & \cdots & E_{M-1,M-1}(z^M) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ z^{-1} \\ \cdots \\ z^{-(M-1)} \end{bmatrix}$$

$$\underline{h}(z) \quad = \quad \underline{E}(z^M) \quad \underline{c}(z). \tag{28}$$

The Type I polyphase representation is used to move the analysis filters after the downsamplers; therefore, the filtering is done more efficiently at the subband data rate, as shown in Fig. 5. $\underline{E}(z)$ is the analysis filter polyphase matrix, and its (k,l) element is

$$E_{kl}(z) = \sum_{n=-\infty}^{\infty} h_k[Mn + l]z^{-n}. \tag{29}$$

The synthesis filter polyphase matrix $\underline{R}(z)$ has similar form.

A PR discrete-time filter bank with the paraunitary property is designed using standard design methods [23]. The paraunitary property is

$$\tilde{\underline{E}}(z)\underline{E}(z) = \alpha I \tag{30}$$

where $\tilde{\underline{E}}(z) \equiv \underline{E}^{\dagger}(1/z^*)$, $\underline{E}^{\dagger}(z)$ is the transpose conjugate of $\underline{E}(z)$, and z^* is the conjugate of z. (Note that the factor α is used in throughout this paper to denote an arbitrary, nonzero

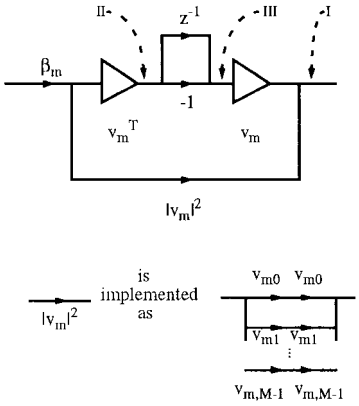

Fig. 6. Implementation of lossless factors $\underline{U}_m(z)$.

scaling factor.) The relationship between the analysis filters and the synthesis filters is

$$\underline{R}(z) = \alpha z^{-N}\underline{E}(z). \tag{31}$$

The filter bank is PR because the transfer function of the filter bank is simply a scaled delay

$$\underline{R}(z)\underline{E}(z) = \alpha z^{-N}\tilde{\underline{E}}(z)\underline{E}(z) = \alpha z^{-N} I. \tag{32}$$

Vaidyanathan develops a lossless factorization of a paraunitary polyphase matrix [24], [25]

$$\underline{E}(z) = \alpha \underline{U}_N(z)\underline{U}_{N-1}(z)\cdots\underline{U}_1(z)\underline{H}_0 \tag{33}$$

where $\underline{H}_0$ is an $M\times M$ unitary matrix ($\underline{H}^{\dagger}{}_0\underline{H}_0 = \alpha I$), and the integer N is determined from the determinant of the polyphase matrix

$$\det\ \underline{E}(z) = \alpha z^{-N} \qquad N \geq 0, \qquad \alpha \neq 0. \tag{34}$$

The lossless factors are iteratively factored out of the polyphase matrix

$$\underline{U}_m(z) = |\underline{v}_m|^2 I - \underline{v}^{\dagger}{}_m + z^{-1}\underline{v}_m\underline{v}^{\dagger}{}_m \tag{35}$$

where the $m\times 1$ vectors $\underline{v}_m$ can be quantized independently to any precision without affecting the lossless property. The lossless factors are implemented with fixed-precision multiplier coefficients. The implementation of each lossless factor is shown in Fig. 6. Note that the factor $|\underline{v}_m|^2$ is implemented as shown

$$|\underline{v}_m|^2 = v_{m0}^2 + v_{m1}^2 + \cdots + v_{m,M-1}^2 \tag{36}$$

since it would normally require higher precision multiplier coefficients, which are not available in some analog implementations. Note that this requires more multiplications, but analog multipliers are relatively small and do not significantly add to the hardware complexity.

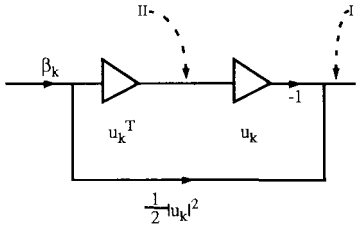

Fig. 7. Implementation of unitary factors $\hat{\underline{G}}_k$.

The unitary matrix $\underline{H}_0$ from (33) is factored with the Householder factorization so that it can be quantized without affecting its unitary property

$$\underline{H}_0 = \alpha\hat{\underline{G}}_1\hat{\underline{G}}_2\cdots\hat{\underline{G}}_{M-1}\hat{\underline{D}} \tag{37}$$

where $\hat{\underline{D}}$ is a constant diagonal matrix. The unitary factors are iteratively factored out of $\underline{H}_0$

$$\hat{\underline{G}}_k = \tfrac{1}{2}|\underline{u}_k|^2 I - \underline{u}_k\underline{u}^{\dagger}{}_k \tag{38}$$

where the $m\times 1$ vectors $\underline{u}_k$ can be quantized independently to any precision without affecting the unitary property. As above, the unitary factors are implemented with fixed-precision multiplier coefficients. The implementation of each unitary Householder factor is shown in Fig. 7. Note that the factor $|\underline{u}_k|^2$ is implemented as above

$$|\underline{u}_k|^2 = u_{k0}^2 + u_{k1}^2 = \cdots + u_{k,M-1}^2 \tag{39}$$

since it would normally require higher precision multiplier coefficients, which are not available in some analog implementations.

The full factorization of the analysis filter polyphase matrix is

$$\underline{E}(z) = \alpha\underline{U}_N(z)\underline{U}_{N-1}(z)\cdots\underline{U}_1(z)\hat{\underline{G}}_1\hat{\underline{G}}_2\cdots\hat{\underline{G}}_{M-1}\hat{\underline{D}}. \tag{40}$$

From (31), the synthesis filter polyphase matrix can be expressed as

$$\underline{R}(z) = \alpha\hat{\underline{D}}\hat{\underline{G}}_{M-1}\hat{\underline{G}}_{M-2}\cdots\hat{\underline{G}}_1\underline{U'}_2(z)\cdots\underline{U'}_N(z) \tag{41}$$

where the lossless factor $\underline{U'}_m(z)$ is defined as

$$\underline{U'}_m(z) = z^{-1}|\underline{v}_m|^2 I + \underline{v}_m\underline{v}^{\dagger}{}_m - z^{-1}\underline{v}_m\underline{v}^{\dagger}{}_m. \tag{42}$$

Note that the vectors $\underline{v}_m$ in $\underline{U'}_m(z)$ and $\underline{u}_k$ in $\hat{\underline{G}}_k$ are identical to those in the analysis filter polyphase factorization so that no further computation is required to calculate the synthesis filter polyphase factorization. Assuming full internal precision arithmetic, the PR property in (32) can be verified for the HFB implemented with this polyphase factorization and quantized (i.e., integer) coefficients.

1) Quantization Algorithm: The following quantization algorithm is used to calculate the n-bit filter coefficients that approximate the infinite-precision filter coefficients specified in the vectors $\underline{v}_m$ and $\underline{u}_k$. In general, a real number y is

quantized to n-bit precision

$$Q(y) = \text{int}_y \times 2^p \tag{43}$$

such that $Q(y)$ is as close to y as possible. The integer int_y is in the range of allowable n-bit integers (e.g., $-2^{n-1} \leq \text{int}_y \leq 2^{n-1} - 1$ for two's complement integers), and the integer p specifies the "binary point." If y is positive, the binary point p is chosen to be the maximum integer such that $y \leq \text{maxval} \times 2^p$, where maxval is the maximum positive integer allowable in n bits (e.g., $2^{n-1} - 1$ for two's complement)

$$p = \left\lceil \log_2 \frac{y}{\text{maxval}} \right\rceil. \tag{44}$$

If y is negative, the binary point p is chosen to be the maximum integer such that $y \geq -\text{minval} \times 2^p$, where minval is the minimum negative integer allowable in n bits (e.g., 2^{n-1} for two's complement)

$$p = \left\lceil \log_2 \frac{y}{\text{minval}} \right\rceil. \tag{45}$$

The vectors $\underline{v}_m$ and $\underline{u}_k$ must be quantized to the same binary point p and should be chosen as in (44) and (45) so that the largest magnitude value in the vectors does not overflow the allowable n-bit integers. Given the binary point p, the integer int_y is rounded to the closest integer to $y/2^p$.

B. Normalization

In the HFB, the implementation of the filters have finite internal precision, which introduces inaccuracies that increase the *distortion* and *aliasing* errors. Normalization factors are used to maintain the highest dynamic range possible, which minimizes the effect of roundoff error and controls the *distortion* and *aliasing* errors to below the desired SFDR of the system. To approximate these finite-precision effects, internal calculations are rounded to an equivalent number of "bits." Without normalization, signals are attenuated through each stage in the factorization, thereby reducing the signal-to-noise ratio and increasing the effect of the round-off error.

To maximize the dynamic range of the system, normalization factors should be calculated to make signal gain as close to but not exceeding unity. The normalization factor must be chosen such that the signal gain can never exceed unity since overflow will occur, causing extreme clipping error in the filter bank.

Fig. 6 shows the three points (I, II, and III) in the lossless factor $\underline{U}_m(z)$ where overflow can possibly occur. The normalization factor β_m before $\underline{U}_m(z)$ should be the largest magnitude possible such that signal gain does not exceed unity at points I, II, or III. The cumulative transfer function for the system from the input to each of these three points is calculated for each successive $\underline{U}_m(z)$ factor. The maximum amplitude signal that can occur at these points occurs when the input signal is such that the coefficients in the cumulative transfer function add coherently. The worst case (maximum gain) of the three check points is noted, and the normalization factor is the inverse. This assures that the worst-case signal does not exceed unity, thereby maximizing the dynamic range of the system while preventing overflow. The three checkpoints for the synthesis filter lossless factors $\underline{U'}_m(z)$, are similar to those for $\underline{U}_m(z)$. Fig. 7 shows the two check points for the unitary Householder factors $\underline{\hat{G}}_k$.

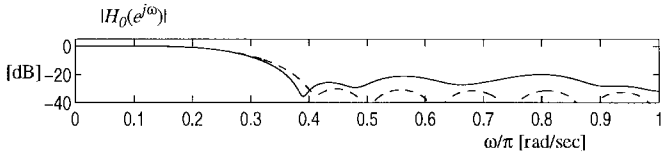

Fig. 8. Analysis filter frequency response with 5-bit coefficients (solid) compared with infinite precision coefficients (dashed).

C. Design Results

An $M = 4$ system based on an FIR length 16, PR cosine-modulated filter bank was designed. The filters were factorized as in Section V-A, and the resulting filter coefficients were quantized to 9 bits (signed binary). Optimal normalization factors were calculated as in Section V-B and quantized to 9 bits. The filter bank with quantized coefficients was verified to be PR. The original cosine-modulated filter bank had −31.5 dB stopband attenuation, and the 9-bit quantization of the coefficient decreased stopband attenuation to only −29 dB. The frequency response is not significantly altered by even extreme quantization; with 5-bit coefficients, stopband attenuation was still better than −20 dB, as shown in Fig. 8.

Simulations were performed on the $M = 4$ system with 9-bit coefficients implemented with finite-precision arithmetic (to approximate a finite-precision sampled-analog implementation). The round-off error does not noticeably affect the frequency response stopband attenuation, but it does increase the *distortion* and *aliasing* errors in the filter bank. With internal precision limited to the equivalent of 15 bits, the maximum *aliasing* error was at −70.5 dB. With internal precision limited to 20 bits, the maximum *aliasing* was −105 dB. Note that a good 12-bit ADC has an SFDR of approximately 80 dB [18]; therefore, the internal precision of a sampled-analog implementation of an HFB should be better than the equivalent of 17 bits so that *aliasing* errors do not limit the resolution of the HFB.

VI. CONCLUSIONS

This paper develops efficient algorithms for the design of M-channel, maximally decimated, uniform bandwidth continuous-time HFB's and discrete-time HFB's. The continuous-time hybrid filter bank analog-to-digital converter employs continuous-time analog analysis filters and discrete-time digital synthesis filters. The filters are designed with enough stopband attenuation to attenuate the effects of mismatches between the converters in the array, which otherwise severely limit the resolution of the system. The filter cutoff should be as sharp as possible to simplify the calibration. This paper develops a discrete-time-to-continuous-time ("Z-to-S") mathematical transform similar to the Pade' approximation, which converts a discrete-time filter to a

continuous-time filter with approximately the same frequency response. This enables PR discrete-time filter banks to be transformed into near-PR continuous-time hybrid filter banks. A stable second-order transform has frequency-response matching better than -28 dB, and a stable fourth-order transform has matching better than -99 dB. Note that a kth-order transform converts an nth-order discrete-time filter into a continuous-time filter with order nk.

With continuous-time analysis filters designed with the Z-to-S transform (or other methods), the discrete-time synthesis filters are designed to minimize the filter bank reconstruction error. An efficient algorithm based upon the fast Fourier transform is developed, and it calculates the optimal real-coefficient FIR filters that minimize the squared reconstruction error in a matter of seconds. A continuous-time HFB with sixth-order analog analysis filters and length 64 FIR synthesis filters with 2.7×10^{-4} dB average deviation from 0 dB *distortion* and -86 dB average *aliasing* error was designed with the Z-to-S transform and the FFT filter design algorithm.

The discrete-time hybrid filter bank analog-to-digital converter employs discrete-time analog analysis filters (such as switched-capacitors or CCD's) and discrete-time digital synthesis filters. A lossless factorization technique is used to design the filters. A paraunitary, PR discrete-time filter bank is factored into lossless factors that can be quantized without affecting the PR property. A gain normalization technique is used to maximize the dynamic range by reducing the effects of round-off error in a finite-precision, sampled-analog implementation. A four-channel discrete-time hybrid filter bank based on a PR, FIR, length 16, cosine-modulated discrete-time filter bank with -31.5 dB stopband attenuation is designed using this lossless factorization. With 9-bit filter coefficients, stopband attenuation was still better than -29 dB. In a sampled-analog implementation with the equivalent of 15 bits internal precision, *aliasing* error was less than -70.5 dB and with the equivalent of 20 bits internal precision, *aliasing* error was less than -105 dB.

Due to the inaccuracies associated with analog implementations, the design of hybrid filter banks is more challenging than for conventional filter banks. The HFB architecture itself is less sensitive to analog mismatch errors such as gain, phase delay, and dc offset in the subband converters. The efficient design algorithms presented in this paper are developed with analog implementation issues at the forefront. With proper calibration, the hybrid filter bank has the potential to overcome analog inaccuracies and provide very high-speed, high-resolution conversion of signals between analog and digital.

REFERENCES

[1] R. E. Crochiere and L. R. Rabiner, *Multirate Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] R. V. Cox, D. E. Boch, K. B. Bauer, J. D. Johnston, and J. H. Snyder, "The analog voice privacy system," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1986, pp. 341–344.

[3] J. Kovacevic, D. J. Le Gall, and M. Vetterli, "Image coding with windowed modulated filter banks," *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, May 1989, pp. 1949–1952.

[4] P. P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 476–492, Apr. 1987.

[5] M. Vetterli, "A theory of multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 356–372, Mar. 1987.

[6] M. J. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 434–441, June 1986.

[7] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*. Cambridge, MA: Wellesley-Cambridge Press, 1996.

[8] S. R. Velazquez, "Hybrid filter banks for analog/digital conversion," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, June 1997.

[9] A. Montijo and K. Rush, "Accuracy in interleaved ADC systems," *Hewlett-Packard J.*, Oct. 1993, pp. 36–42.

[10] A. Petraglia and S. K. Mitra, "Analysis of mismatch effects among A/D converters in a time-interleaved waveform digitizer," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 831–835, Oct. 1991.

[11] R. Khoini-Poorfard, "Mismatch effects in time-interleaved oversampling converters," *Proc. IEEE. Int. Symp. Circuits Syst.*, London, U.K., May 1994, pp. 429–432.

[12] A. Petraglia and S. K. Mitra, "High speed A/D conversion using QMF banks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1990, pp. 2797–2800.

[13] *National Semiconductor Data Acquisition Databook*, National Semiconductor, 1993.

[14] H. Shu, T. Chen, and B. A. Francis, "Minimax design of hybrid multirate filter banks," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 120–128, Feb. 1997.

[15] S. Paul, *Analysis, Design, and Implementation of Charge-to-Digital Converters*, Master's thesis, Mass. Inst. Technol., Cambridge, May 1995.

[16] B. Brannon, "Wide-dynamic-range A/D converters pave the way for wideband digital-radio receivers," *Electron. Des. News*, Nov. 7, 1996, pp. 187–205.

[17] *National Semiconductor 1996 Databook*, "Comlinear CLC952 12-bit 41 MSPS monolithic A/D converter," National Semiconductor, Aug. 1996.

[18] *Analog Devices Databook*, "12-Bit, 41 MSPS monolithic A/D converter AD9042," Analog Devices, 1996.

[19] S. R. Velazquez, T. Q. Nguyen, S. R. Broadstone, and J. K. Roberge, "A hybrid filter bank approach to analog-to-digital conversion," in *Proc. IEEE-SP Int. Symp. Time-Frequency Time-Scale Anal.*, Oct. 1994, pp. 116–119.

[20] S. R. Velazquez, "A hybrid quadrature filter bank approach to analog-to-digital conversion," Master's thesis, Mass. Inst. Technol., Cambridge, June 1994.

[21] A. V. Oppenheim and R. S. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[22] G. A. Baker, Jr., *Essentials of the Padé Approximation*. New York: Academic, 1975.

[23] T. Q. Nguyen and R. D. Koilpillai, "The theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect-reconstruction," *IEEE Trans. Signal Processing*, vol. 44, pp. 473–483, Mar. 1996.

[24] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[25] P. P. Vaidyanathan, T. Q. Nguyen, Z. Doganata, and T. Saramaki, "Improved technique for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1042–1056, July 1989.

Scott R. Velazquez received the B.S. degree in electrical science and engineering and the M.S. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1994. He received the electrical engineer degree from MIT in 1996 and the Doctor of Philosophy degree in electrical engineering from MIT in 1997. He was an NSF Graduate Fellow in Electrical Engineering at MIT while doing his doctoral work.

He participated in the VI-A internship program at MIT Lincoln Laboratory (Group 44, Advanced Techniques) for his undergraduate and graduate work. He also performed research in the MIT Strobe Laboratory and the MIT Research Laboratory of Electronics. He is currently President of The V Company, Revere Beach, MA. His research interests include mixed analog/digital signal processing and wireless communications.

Dr. Velazquez received an NSF Incentives for Excellence Prize in 1991, an NCR Scholarship from 1989 to 1993, and a Northern Telecom Electronics Scholarship in 1989.

Truong Q. Nguyen (S'85–M'90–SM'95) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1985, 1986, and 1989, respectively.

He was with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington, MA, from June 1989 to July 1994 as a Member of Technical Staff. From 1993 to 1994, he was a visiting lecturer at MIT and an adjunct professor at Northeastern University, Boston, MA. From August 1994 to July 1996, he was an Assistant Professor at the University of Wisconsin, Madison. He is now with Boston University, Boston, MA. His research interests are in digital and image signal processing, multirate systems, wavelets and applications, and biomedical signal processing.

Prof. Nguyen was a recipient of a fellowship from Aerojet Dynamics for advanced studies. He received the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (Image and Multidimensional Processing area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks in 1992. He received the NSF Center Award in 1995 and is the coauthor (with Prof. G. Strang) of the textbook *Wavelets and Filter Banks* (Cambridge, MA: Cambridge Wellesley). He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He has also served in the DSP Technical Committee for the CAS society. He is a member of Tau Beta Pi and Eta Kappa Nu.

Steven R. Broadstone received the B.A. degrees in mathematics and physics from Hastings College, Hastings, NE, in 1982 and the B.S., M.S., and D.Sc. degrees in electrical engineering from Washington University, St. Louis, MO, in 1984, 1986, and 1988, respectively.

He is currently the Vice President, Research and Development, at Teratech Corporation, Burlington, MA, specializing in the development of state-of-the-art medical ultrasound systems. He is also involved in the advancement of testing methodologies and standards for analog-to-digital converters as co-editor of the IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters. Before that, he was involved in the development of adaptive-array processing algorithms and systems at the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington, MA. While at Washington University, he was a research assistant at the Biomedical Computer Laboratory and the Electronic Systems and Signals Research Laboratory (ESSRL) in the Department of Electrical Engineering, pursuing research in the application of digital-signal processing technology to real-time medical imaging using ultrasound. He has published papers and technical reports in the areas of adaptive-array processing, analog-to-digital converter evaluation, and medical ultrasonic imaging.

Dr. Broadstone is a member of the IEEE Instrumentation and Measurement Society's TC-10 subcommittees on Waveform Analyzer Testing Standards and Analog-to-Digital Converter Testing Standards and has served as a reviewer for the IEEE TRANSACTIONS ON ANTENNAS ON PROPAGATION, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, and the IEEE TRANSACTIONS ON MEDICAL IMAGING.