A MULTISTAGE FILTERBANK-BASED CHANNELIZER AND ITS MULTIPLIER-LESS REALIZATION

C. Y. Fung and S. C. Chan
Department of Electrical and Electronic Engineering
The University of Hong Kong

ABSTRACT

This paper proposes a multistage filterbank-based channelizer for software radio base stations. The proposed channelizer is capable to receive channels with potentially different bandwidths as required in a multi-standard cellular based station. It consists of multiple stages of DFT filter banks and efficient sample rate changers. The front-end DFT filter bank of the channelizer has a fixed number of channels but the passband supports overlap with each other. The received signals selected by a given output of this filter bank are fed into sample rate changers so that they can fit into the fixed channel spacing of the DFT filter banks in the following stages. Due to the lowered sample rate, these back-end DFT filter banks can have either fixed or variable number of channels. Repeatedly using this multistage architecture, channels with different bandwidths can be isolated. The design and implementation of the proposed channelizer are discussed in detail. An example of dual-mode GSM/W-CDMA channelizer is also discussed to illustrate the proposed design methodology.

I. INTRODUCTION

Software radio is a general hardware/software platform for supporting the communication between different wireless communications systems. It is potentially a cost-effective mean for migration to future wireless communication standards [1,2]. Software radio base station, in particular, is very attractive in that it allows rapid deployment of new services with simple upgrade of software modules.

Due to current limitation of analog-to-digital converter (ADC) and digital signal processor (DSP), the received radio signal is usually digitized at the intermediate frequency (IF). A flexible IF processing architecture for isolating the user's channel is an important issue of software radio. In a base station employing the software radio concept, an IF channelizer is used to extract different radio channels from the IF spectrum, and down-convert them back to baseband, at a lower sample rate suitable for baseband processing. Efficient implementations of channelizer have been proposed using the Discrete Fourier Transform (DFT) filter bank [3,4]. The basic principle of the DFT filterbank-based channelizer is to separate simultaneously a number of equally spaced channels using a uniform filter bank derived from the modulation of a given prototype filter. The complexity of the channelizer is therefore greatly reduced compared with traditional analog channelizer. Unfortunately, the channel bandwidth of a DFT filter bank is fixed for a given sample rate and channel number of the filter bank. In other words, once it was built, it can only support a given air interface standard. Many individual channelizers are thus required to support other wireless communications standards in one single cellular base station. This motivates us to investigate new channelizer architecture based on multistage DFT filter banks and efficient sample rate changers (SRC), which is more flexible in handling signals with different bandwidths. More precisely, the DFT filter bank at the front end of the channelizer has a fixed number of channels but their frequency supports overlap each other considerably. A SRC is then used to convert their sample rate so that they can fit into that of the DFT filter banks in the following stages, which can have a fixed or variable but larger number of channels. Repeatedly using this multistage architecture, channels with different bandwidths can be isolated.

II. CONVENTIONAL FILTERBANK CHANNELIZER

The architecture of a uniform DFT filter bank using polyphase decomposition is shown in Fig. 1. It consists of a set of bandpass filters $h_k[n]$, for $k = 0,\ldots K-1$, equally spaced at center frequency $\omega_i = 2\pi i / K$ where K is the number of channels in the channelizer. In the DFT filter bank, $h_k[n]$ is obtained by modulating a low-pass prototype filter $h_o[n]$ of length N with real coefficients using the inverse DFT (IDFT). The Fourier transform of $h_o[n]$ is given by:

$$H_o(z) = \sum_{l=0}^{K-1} z^{-l} E_l(z^K), \text{ where } E_l(z^K) = \sum_{n=0}^{P-1} h_o[nK+l] z^{-nK} \quad (1)$$

and $E_l(z)$ is the lth polyphase filter and the number of taps per phase P is N/K. For critical sampling, the number of channels K and the downsampling ratio M are equal. When they are not equal, the filter outputs are oversampled by a factor

$$T = K / M . \quad (2)$$

In order to avoid aliasing, the passband edge ω_p and stopband edge ω_s of the prototype filter should satisfy [5]:

$$\omega_p = \pi / K \text{ and } \omega_s \le 2\pi / K , \quad (3)$$

and the center frequency of the kth channel is given by:

$$\omega_k = 2\pi k / K . \quad (4)$$

Usually ω_k is chosen as the channel spacing f_{cs} of the received channels. If the sample rate of the ADC is F_s, then

$$F_s = K \times f_{cs} \quad (5)$$

and the sample rate of the output signal in the DFT filter bank channelizer f_s is given by:

$$f_s = F_s / M . \quad (6)$$

In order to interface the channelizer directly to the baseband processing unit, f_s must be a multiple of the baud rate f_b. Since both K and M are integers, a SRC is usually required to fractionally decimate the output signal of the channelizer to a rate that is a multiple of the baud rate for baseband processing. If

0-7803-7448-7/02/$17.00 ©2002 IEEE

F_s is the least common multiple (LCM) of f_s and f_{cs}, then the SRC can be eliminated.

The baud rates f_b for different wireless communication systems, however, differ from each other and the conventional DFT filter bank architecture is not able to handle channels with different bandwidths. The proposed multistage architecture, on the other hand, divides the input spectrum successively using a set of DFT filter bank channelizers with different number of channels in each stage so that channels with decreasing bandwidths can be isolated successively. In the following section, the design methodology of the multistage filter bank (MSFB) channelizer will be explained in more detail.

III. MULTISTAGE FILTERBANK CHANNELIZER

Without loss of generality, a two-stage MSFB channnelizer is considered in this paper and is illustrated in Fig. 2. It can be extended to have multiple stages without much difficulty. The front-end DFT filter bank has a fixed number of channels, but the passband supports overlap with each other considerably so that the channels midway between the center frequencies of successive bandpass filters in the filter bank can be isolated. A SRC is then used to fractionally decimate the outputs of the front-end DFT filter bank. Each of them can then be fitted into a uniform channel in the back-end DFT filter bank in the following stage, which usually has a larger number of channels. Finally, the outputs of the back-end channelizers are fed to the baseband processing units. Additional sample rate changing maybe necessary which can be implemented by a DSP since the sample rate is significantly lowered.

The main reason for using a front-end DFT filter bank with fixed number of channels is that it can be efficiently implemented with less hardware resources and hence lower power consumption, using the canonical signed digits (CSD) representation of the fixed coefficients. It can then be implemented at high-speed using say Application-Specific Integrated Circuit (ASIC) chips. The number of channels in the back-end channelizer and the conversion ratio of the SRC can be variable because the front-end channelizer has already decimated the IF signal to a relatively low sample rate. Therefore the SRC and back-end channelizer can be implemented using either ASIC or high-performance DSP. The multistage architecture allows one to perform comfortably the tradeoff between system flexibility, hardware cost and power consumption, apart from being able to isolate receiving channels with different bandwidths. Its complexity is only slightly higher than the conventional approach, with additional computational load mainly due to the SRC.

A. Front-end and Back-end Channelizers

The structure of the front-end and back-end channelizers of the MSFB channelizer is similar to the DFT filter bank channelizer described in section II. Assume that the number of channels in the front-end and back-end channelizers are K_1 and K_2 respectively. The maximum number of radio channels that can be extracted by the MSFB channelizer K is given by:

$$K = K_1 \cdot K_2 . \tag{7}$$

Both K_1 and K_2 are chosen to be a power of 2 so that the IDFT can be implemented with efficient radix-2/radix-4 FFT algorithms. Both the front-end channelizer and the back-end channelizer are designed as oversampled DFT filter bank instead of critically sampled so that the passband of the kth channel slightly overlaps with that of the $(k-1)$th and $(k+1)$th channels. Unlike the design in section II, the passband edge ω_p and the stopband edge ω_s of the low-pass prototype filter are given by

$$\omega_p = (1+d)\pi / K \quad \text{and} \quad \omega_s = 2\pi / K , \tag{8}$$

where d is the overlapping factor with a value between 0 and 1. The center frequency ω_k of each channel is chosen as a multiple of the largest channel separation of the standards supported. Multiples of these channels will fall into the passband of the bandpass filters and are extracted by the channelizer. To extract other possible channels in between, d should be chosen as:

$$d = K \cdot f_{cs}' / F_s , \tag{9}$$

where f_{cs}' is the second largest channel separation among the wireless standards to be supported by the base station. Fig. 3 shows the magnitude response of the proposed filter bank. The shaded region denotes the frequencies overlapped by successive bandpass filters and the black shaded region denotes other possible received radio signals. Hence the received radio signal can be extracted by the filter bank even its frequency is midway between the center frequencies of successive bandpass filters of the filter bank. The unwanted signals with frequencies from ω_p to $2\pi / K$ are filtered partially by the bandpass filters of the filter bank and will be removed by the anti-aliasing filters of the SRC in the later stage.

Given the maximum passband ripple δ_p and the maximum stopband ripple δ_s, the length of the prototype filter can be estimated using the Remez-exchange algorithm [6]. Let δ_{po} and δ_{so} be the overall required peak passband and stopband ripples of the MSFB channelizer. The magnitude response of the prototype filter in front-end or back-end channelizer can be chosen as:

$$\delta_p \cong \delta_{po} / 2 \quad \text{and} \quad \delta_s \cong \delta_{so} . \tag{10}$$

B. Sample Rate Changer

In this section, we briefly outline the design of the programmable SRC in the MSFB. In general, there are two approaches to implement a programmable SRC. One is to store a set of impulse response of the fractional delay digital filter required to interpolate the bandlimited signal at the desired sampling instants. This requires extensive computational load and thus not suitable for high-speed applications. Another type of implementation is shown in Fig. 4. A programmable sample rate changer similar to that proposed in [7] is used to perform arbitrary sample rate conversion. The input signal is first upsampled by a factor of L. The L-band filter can be implemented as a cascade of half-band filter if L is a power of two. Alternatively, prefilter such as the cascaded integrator-comb (CIC) filter can be employed for large decimation factor. If L is sufficiently large, further interpolation by an irrational number can be achieved simply by a second or higher order polynomial interpolation such as Lagrange interpolation or cubic

spline [8]. The Farrow structure [9], which is usually used to realize tunable fractional delay digital filter, can also be used to implement the polynomial interpolator. Whilst the Farrow structure can be efficiently implemented using multiplier-block exists as described in [10].

Because of the non-ideal response of the low order interpolation (decimation), the upsampling ratio has to be sufficiently large, depending on the required signal-to-distortion ratio (SDR). For second order interpolation, the signal-to-distortion ratio of the linear interpolation process is given by [11]:

$$\text{SDR} \geq 80 \cdot U^4 / w_x^4, \tag{11}$$

where U is the number of subfilters required (upsampling ratio) and w_x is the bandwidth of the input signal from the front-end channelizer. From equations (2) and (8), the bandwidth of the input signal to the SRC depends on the oversampling ratio and the stopband cutoff frequencies of the low-pass filter of the front-end channelizer. For a front-end channelizer with lower oversampling ratio, more subfilters are needed for the SRC due to the wider bandwidth of the input signal to the SRC, and vice versa. Hence, to achieve a given minimum SDR, there exists a trade-off between the complexity of the front-end channelizer and the SRC.

Let the decimation ratios of the front-end and back-end channelizers be M_1 and M_2, respectively. If the frequency of the digitized signal from the ADC is given by F_s, then the sample rate of the MSFB channelizer's output f_s is given by

$$f_s = F_s \cdot M_{src} (M_1 M_2)^{-1}, \tag{12}$$

where M_{src} is the conversion ratio of the SRC.

To efficiently implement the MSFB channelizer, f_s can be chosen to be a non-integer multiple of the baud rate. Instead, an additional SRC is used to convert f_s into a multiple of f_b. Fortunately, since f_s is now at a relatively low frequency, the additional SRC can be embedded into the DSP processors that perform the baseband demodulation process.

IV. DUAL-MODE GSM/W-CDMA CHANNELIZER

In this section, an example of dual-mode GSM/W-CDMA receiver is given to illustrate the design methodology of the MSFB channelizer. The architecture of the proposed dual-mode IF channelizer *for receiving a single channel* is illustrated in Fig. 5. Some essential specifications of GSM and W-CDMA are shown in Table 1.

Assume that the sample rate of the ADC is 80 Msps and the dynamic range of the ADC is sufficient to comply with all the standards. The digital channelized receiver is capable of receiving an IF signal of 40MHz real bandwidth. According to equation (8) and (9), the passband and the stopband cutoff frequencies of the prototype filter in the front-end channelizer are chosen as $(1.04)\pi / K_1$ and $2\pi / K_1$, respectively. Table 2 shows the parameters of the MSFB channelizer. For the back-end channelizer, since no overlapping is necessary, the passband and the stopband cutoff frequencies of the prototype filter are chosen as π / K_2 and $2\pi / K_2$, respectively. Typical values of the peak passband and stopband ripples are used. As an example, the magnitude response of the front-end channelizer is shown in Fig. 6. The front-end channelizer has a downsampling ratio of 2, so that its output has a sample rate of 40 Msps. For W-CDMA signal, the output of the front-end channelizer is decimated to the baud rate directly. For GSM signal, the output signal of the front-end channelizer is decimated to 6.4 Msps by the SRC and further divided by the back-end channelizer.

The design of SRC for W-CDMA and GSM reception is summarized in Table 3. Note that back-end channelizer is not required for the reception of W-CDMA signals and the desired signals are obtained by sample rate conversion of outputs of front-end channelizer only. Second order interpolation is chosen in this example due to its low complexity yet high SDR for low upsampling ratio.

	GSM	W-CDMA
Channel separation f_{cs}	200 kHz	5 MHz
Baud rate f_b	270.8 ksps	3.84 Msps
Number of channels (F_s / f_{cs})	400	16

Table 1. Requirements for the channelizer.

	Front-end	Back-end
Number of channels	16	32
Oversampling ratio T	8	4
Prototype filter length N	144	256
Passband ripple	$5\text{x}10^{-4}$	
Stopband attenuation	10^{-4}	

Table 2. Parameters of the MSFB channelizer.

	GSM	W-CDMA
Conversion ratio M_{src}	4/25	12/125
$w_x \approx 2\pi f_b / f_s$	0.0135π	0.192π
Number of subfilters U	1	4
Signal-to-distortion ratio	10^5	
Interpolation	2^{nd} order (linear interpolation)	

Table 3. Parameters of the SRC.

V. MULTIPLIER-LESS FILTER BANKS

We now briefly describe the implementation of the proposed MSFB channelizer. To reduce the complexity of the channelizer, the polyphase filter in the channelizer can be implemented by representing the filter coefficients in Sum-of-Powers-of-Two (SOPOT) coefficients or CSD. Thus, coefficients multiplications can be implemented by limited number of shifts and additions only. Furthermore, an efficient technique called multiplier-block [10,12] can be used to realize the multiplier-less filter bank with minimum number of adders. In the latter case, one must operate the polyphase filter at a higher operating frequency in this approach. The area of the channelizer is greatly reduced at the expense of higher power dissipation. Due to page limitation, we only describe the use of multiplier block to implement the prototype filters of the MSFB, although it is also applicable to the implementation of SRC. More details about the efficient implementation of the entire MSFB using multiplier block can be found in [13]. Assume that each SOPOT coefficient of the low-pass filter is represented by a 16-bit CSD number, the performance of multiplier block is summarized in Table 4.

	Without Multiplier Block	With Multiplier Block
Number of multiplier	4240	0
Number of adders	4223	1283

Table 4. Performance of the multiplier block.

VI. CONCULSION

A new multistage filterbank-based channelizer for software radio base stations is presented. Unlike conventional uniform DFT filterbank-based channelizer, it is more flexible to handle receiving channels with potentially different bandwidths as required in a multi-standard cellular based station. An example dual-mode GSM/W-CDMA channelizer is also discussed to illustrate the proposed design methodology. Implementation issues of the SRC in the multistage channelizer are discussed. Efficient multiplier-less realization of the MSFB channleizer is also given.

REFERENCES

[1] J. Mitola, "Software Radio Architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26-38, May 1995.

[2] E. Buracchini, "The Software Radio Concept," *IEEE Commun. Mag.*, vol. 38, no. 9, pp. 138-143, Sep. 2000.

[3] K. C. Zangi and R. Koilpillai, "Software Radio Issues in Cellular Base Stations," *IEEE J. Select. Areas Commun.*, vol.17, no. 4, pp. 561-573, Apr. 1999.

[4] D. R. Zahirniak, D. L. Sharpin and T. W. Fields, "A Hardware-efficient, Multirate, Digital Channelized Receiver Architecture," *IEEE Trans. Aerospace and Electronic Systems*, vol. 34, no. 1, Jan. 1998.

[5] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall, 1983.

[6] S. K. Mitra and J. F. Kaiser, *Handbook for Digital Signal Processing*, New York: Wiley, 1993.

[7] T. Saramaki and T. Ritoniemi, "An Efficient Approach for Conversion between Arbitrary Sampling Frequencies," in *Proc. ISCAS'96*, Atlanta, GA, May 1996.

[8] T. A. Ramstad, "Fractional Rate Decimator and Interpolator Design," in *Proc. EUSIPCO' 98*, Rhodes, Greece, Sep. 1998.

[9] C. W. Farrow, "A Continuously Variable Digital Delay Element," in *Proc. ISCAS'88*, Espoo, Finland, Jun. 1988.

[10] C. K. S. Pun, Y. C. Wan, S. C. Chan and K. L. Ho, "An Efficient Design of Fractional-Delay Digital FIR Filters using the Farrow Structure," in *Proc. Workshop on SSP*, Singapore, Aug. 2001.

[11] J. G. Proakis, C. M. Radar, F. Ling and C. L. Nikias, *Advanced Digital Signal Processing*, New York: Macmillan, 1992.

[12] A.G. Dempster and M.D MacLeod, "Use of multiplier blocks to reduce filter complexity," in *Proc. ISCAS'94*, London, May 1994.

[13] K. S. Yeung and S. C. Chan, "On The Design and Multiplier-less Realization of Digital IF for Software Radio Receiver," submitted to *EUSIPCO 2002.*

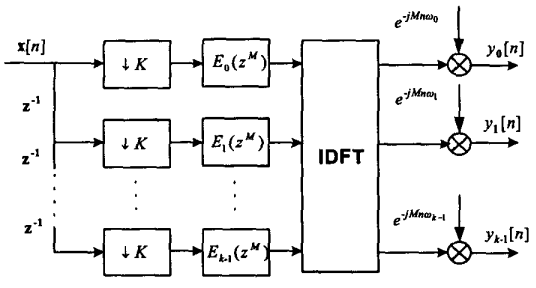

Fig. 1. DFT filter bank channelizer with polyphase decomposition.

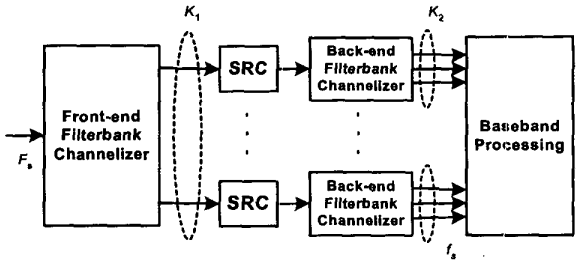

Fig. 2. Architecture of the MSFB channelizer.

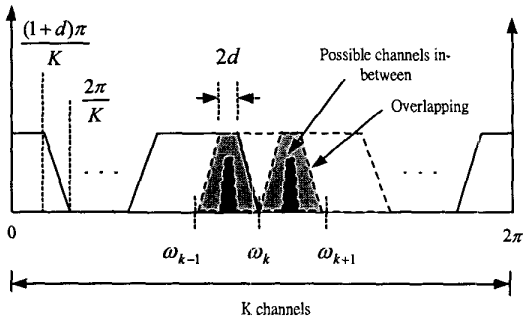

Fig. 3. Magnitude response of the front-end DFT filter bank.

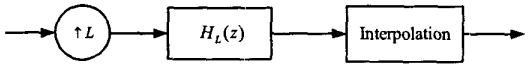

Fig. 4. Programmable sample rate changer.

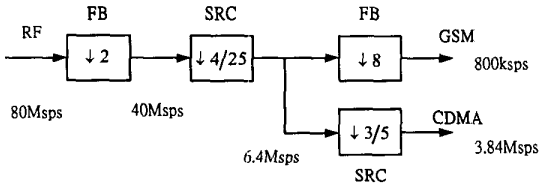

Fig. 5. Architecture of IF processing.

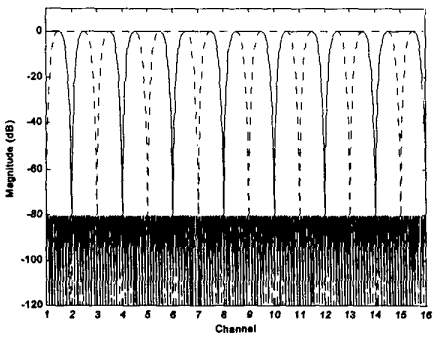

Fig. 6. Magnitude response of the front-end channelizer