

Module 9 AUDIO CODING

Version 2 ECE IIT, Kharagpur

Lesson 30 Polyphase filter implementation

Instructional Objectives

At the end of this lesson, the students should be able to :

1. Show how a bank of bandpass filters can be realized through windowing and transform.
2. Justify why the filters implemented as above are known as polyphase filters
3. Describe the polyphase implementation of the analysis filter bank-in MPEG-1 audio
4. Describe the polyphase implementation of the synthesis filter bank in MPEG-1 audio.

30.0 Introduction

In lesson 29, we had presented two approaches to analysis synthesis filter design. One is through a direct implementation of filter banks in time domain (through FIR filters) with overlapped frequency domain characteristics. Such filters require frequency domain alias correction by the proper design of adjacent filter bank characteristics. The second method windows the input samples and transforms those through DCT/DST. These filters require time domain alias correction, which we described in the previous lesson.

We had also stated that the second approach leads to a better efficiency and has been adopted in the MPEG-1 audio coding standard. Although we had mentioned that a duality that exists between the time domain and frequency domain implementation, in this we shall analytically establish this duality for a better understanding of the polyphase analysis and synthesis filters . In this lesson, we are also going to present the implementation of these filters, as given in MPEG-1 standard.

30.1 Theory of Polyphase filters

We have discussed in the previous lesson that the design of a multiband filter requires alias cancellation between adjacent bands and that the filter shapes be controlled such that the transition bands of adjacent filters add to produce a flat response.

Following Rothwieter's approach [1], a filter bank fulfilling these requirements can be formed by first designing a low pass prototype filter with a controlled transition band frequency response. The filter bank can then be composed by multiplying the impulse response of the prototype low pass filter with a sinusoid having frequencies equal to the centre frequencies of the desired filters.

We assume the sampling frequency to be unity, so that the frequency range to be covered is 0 to π radians / sec. A bank of M filters to be synthesized as shown in fig 30.1. The nominal bandwidth of each filter is π/ M and the filter centre frequencies are at add multiples of $\pi/2M$.

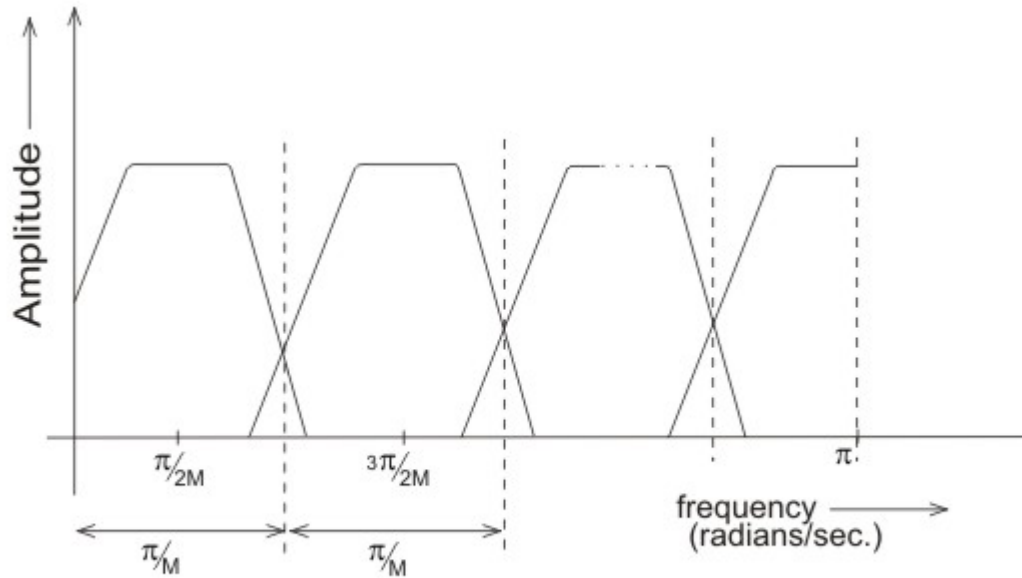


FIGURE 30.1 A bank of bandpass filter. Sampling frequency is assumed to be 1 Hz.

Real bandpass filters are composed of two complex filters $F_i(z)$ and $G_i(z)$ located respectively at the positive and negative centre frequencies. If the prototype lowpass filter is a FIR filter with impulse response $h(n)$ and z- transform $H(z)$ we can express $F_i(z)$ and $G_i(z)$ as

$$\begin{aligned} F_i(z) &= H\left(e^{-\frac{j\pi(2i+1)}{2M}} z\right) \\ G_i(z) &= H\left(e^{\frac{j\pi(2i+1)}{2M}} z\right) \end{aligned} \quad \dots\dots\dots(30.1)$$

In equations (30.1), the complex filters in z-domain are derived by multiplying the Z with complex exponentials located of odd multiples of $\pi/2M$. We therefore obtain a band of filters with $i=0,1,\dots,M-1$ and the corresponding composite filters may be expressed as

$$\begin{aligned} H_i(z) &= a_i F_i(z) + b_i G_i(z) \\ K_i(z) &= c_i F_i(z) + d_i G_i(z) \end{aligned} \quad \dots\dots\dots(30.2)$$

where a_i , b_i , c_i and d_i are complex constants. $H_i(Z)$ and $K_i(z)$ form the analysis and the synthesis filter banks respectively, as shown in fig 30.2

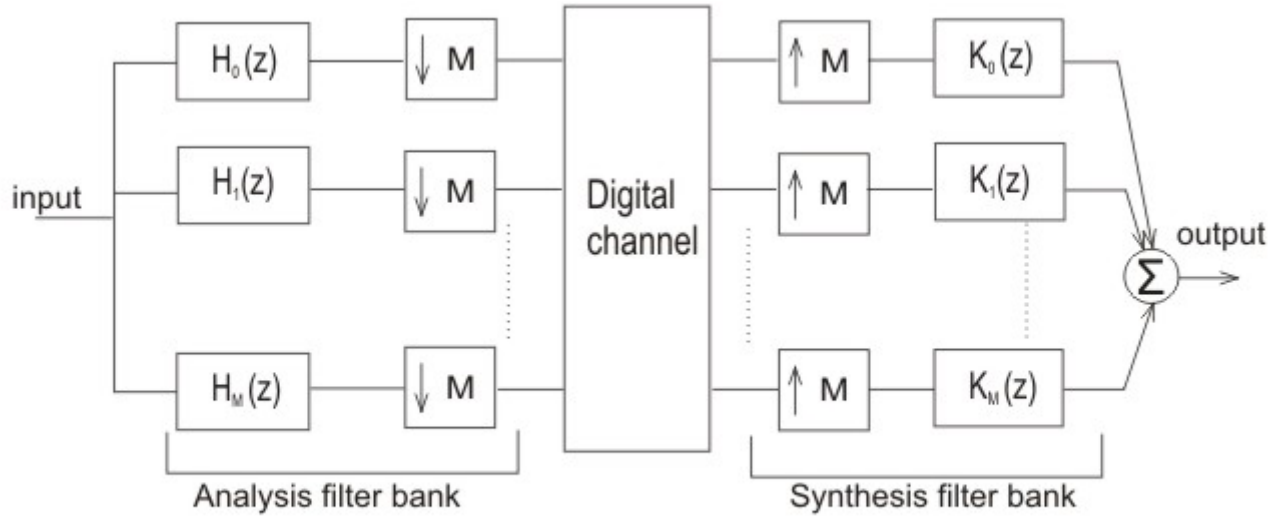


FIGURE 30.2 Analysis and Synthesis filter banks

If $a_i = b_i^*$, the impulse response of H_i is real and is given by

$$h_i(n) = \begin{cases} \text{Re}(a_i) \cos \left[\pi (2i+1)(2n+1)/4M \right] \\ -\text{Im}(a_i) \sin \left[\pi (2i+1)(2n+1)/4M \right] \end{cases} h(n) \dots \dots \dots (30.3)$$

Replacing a_i and b_i with c_i and d_i , a corresponding equation results for $k_i(n)$.

From equation (30.3), it is evident that the $h_i(n)$ contains odd number of half cycles of sinusoids in $2M$ points ($n=0, 1, \dots, 2M-1$)

If the input samples are given by $x(n)$, $n=0, 1, \dots$, then the filtered output of bank- i is given by the famous convolution equation

$$s_i(i) = \sum_{n=0}^{L-1} x(l-n)h_i(n) \dots \dots \dots (30.4)$$

where L is the length of the filter tap.

Substituting equation (30.3) in (30.4) we obtain DCT/DST of the input samples, multiplied by the prototype low pass filter's impulse. The latter serves as the windowing of the time domain samples.

The equation (30.3) involving $h_i(n)$ clearly shows that the response of each band i is a modulation of the prototype response with a cosine term to shift the low pass response to the appropriate band. Hence, these are called polyphase filters.

One efficient implementation of a polyphase filter bank is shown in fig.30.3. The input samples are first multiplied by lowpass prototype filter $h(n)$. Blocks of $2M$ products of the multiplications are accumulated with the sign of alternate blocks negated. These $2M$ values are then multiplied by M sinusoids to generate the M output values. A typical response of low-pass prototype analysis filter $h(n)$ is sketched in fig 30.4. Negating the signs of alternate blocks result in a net windowing function $c(n)$ as sketched in fig 30.5.

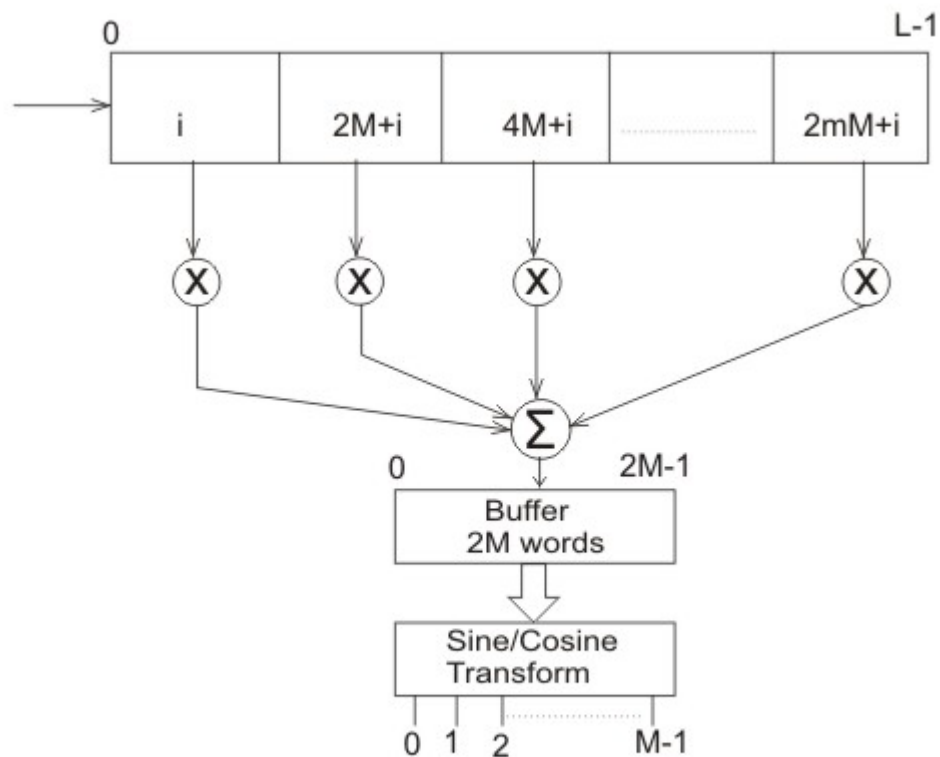


FIGURE 30.3 Implementation of polyphase filter

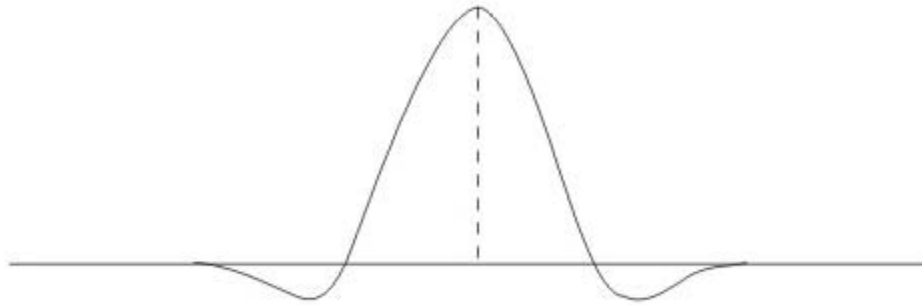


FIGURE 30.4 A plot of $b(n)$

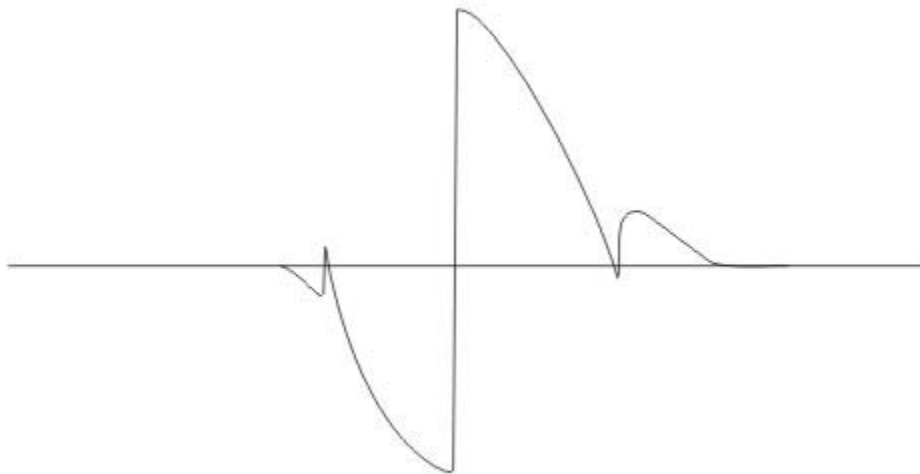


FIGURE 30.5 A plot of $c(n)$

For audio signals, since the flat response to DC is not required, the coefficient a_i in equation (30.3) can be either pure real or pure imaginary. Hence, equation (30.3) contains either the sine terms or the cosine terms, but not both. In MPEG-1, a modified discrete cosine transforms (MDCT) is applied to obtain M subband samples out of $2M$ words

30.2 Polyphase analysis filter for MPEG-1 audio

Fig.30.6 shows the polyphase implementation of MPEG-1 analysis filter bank which is a variant of Rothweller's implementation described above. During the analysis, the audio signal is shifted into a 512 samples \times buffer, 32 samples at a time.

Analysis Filterbank

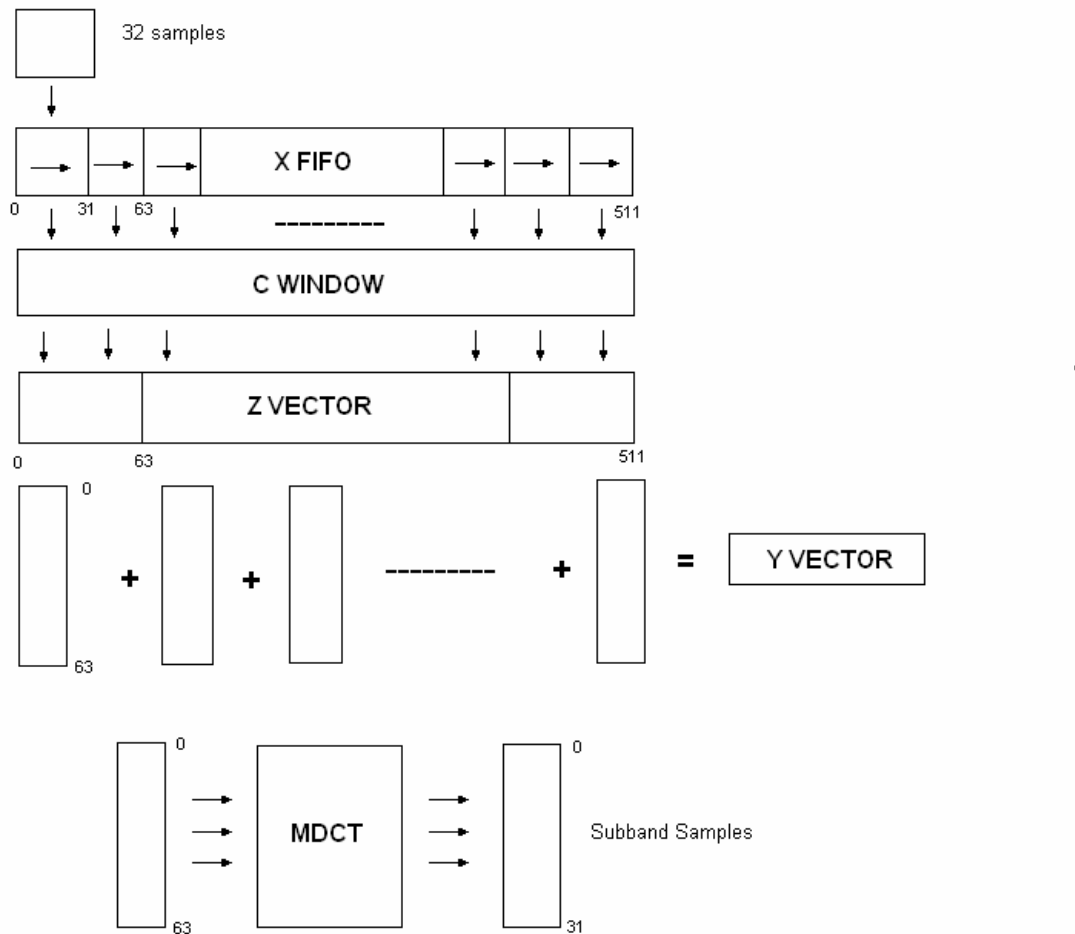


Fig.30.6 Polyphase implementation of analysis filterbank.

The content of X buffer are multiplied by the C-window function $c(n)$ and the results are stored into the Z-buffer. The Z-buffer contents are divided into eight 64-element vectors (taking $M=32$), which are summed to form a 64-element Y-vector. The Y-vector is transformed using MDCT to yield the 32-subband samples.

30.3 Polyphase synthesis filter for MPEG-1 audio

The synthesis filter shown in fig 30.7 performs exact reversal of the operations performed by the analysis filter. The 32 subband samples are transformed back to the 64 element V vector, using inverse MDCT (IMDCT). The V-vector is pushed into a FIFO which stores the last 16 V vectors. A U-vector is created from the alternate 32 component blocks and a window (called D-window) is applied to U to produce the W-vector, which is divided into 16 vectors, each having 32

values. These 16 vectors are added together to obtain 32 sample output. It may therefore be noted that the analysis synthesis filtering operation is critically sampled, since exactly 32-sampled output is obtained for 32-sampled input.

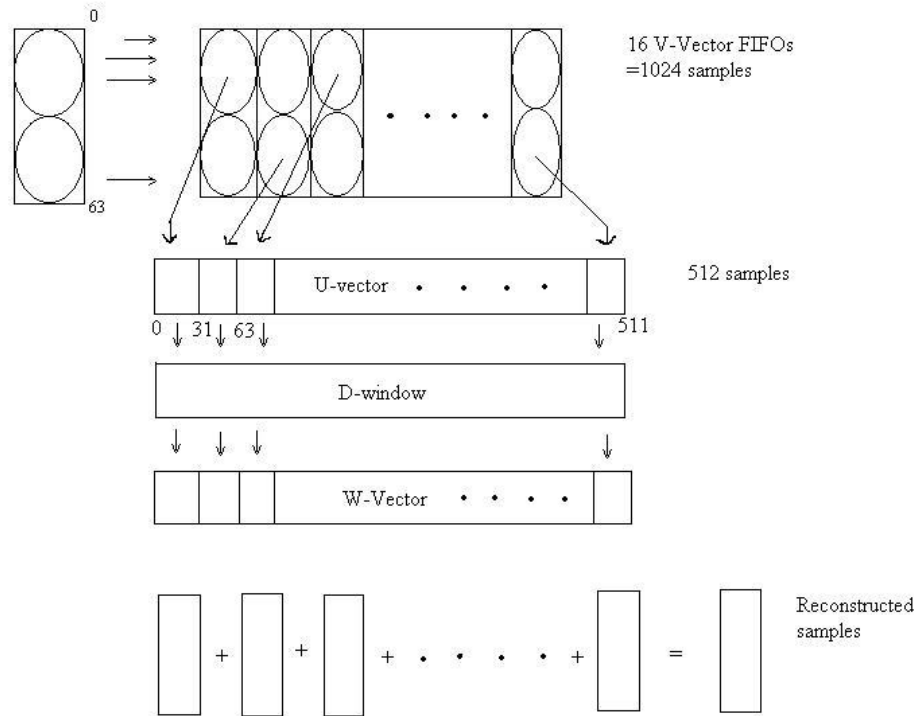


Fig 30.7 Synthesis filter bank of MPEG-1

30.4 Conclusion

In this lesson, we have learnt how a multi band filter can be implemented from a low-pass prototype filter using a polyphase approach, by modulating the low pass filter response with sinusoids having frequencies corresponding to the centre frequencies of the filter bank. The filter bank can be realized by windowing the samples, followed by a sinusoidal transformation. We are yet to learn how the subband samples are encoded to achieve audio compression. This will be our focus in lesson-31.

References

1. J.H. Rothweller " Polyphase Quadrature Filters – A new subband coding technique", International Conference on Acoustic, Speech and Signal Processing (ICASSP), Boston, 1983, pp 1280-1283.
2. S. Shlien, "Guide to MPEG-4 Audio Standard" IEEE Transaction on Broadcasting, Vol. 40, No.4, December, 1994, pp 206-218.

