WAVELETS AND MULTIRATE DIGITAL SIGNAL PROCESSING

Lecture 35: The Lifting Structures And Polyphase Matrices

Prof. V.M.Gadre, EE, IIT Bombay

1 Introduction

In the last lecture we derived a computationally efficient structure to realize orthogonal filter banks, this structures are called as lattice.
Lattice is a periodic repetition of uniform modular piece. We also saw that complexity was more on the lattice structure as it simultaneously work on two inputs to give two outputs.
For computationally efficient realization of orthogonal filter banks we will go for further simplification of lattice structure *i.e.*, we are going to decompose lattice stages into two sub stages which have more elementary operation that is called as lifting structure and this will lead to the idea of polyphase matrices.

2 The lifting structures and polyphase matrices

Figure 1 shows a simple lattice stage.
Here K is known as lattice parameter. It distinguishes one stage from other. From Figure 1 it

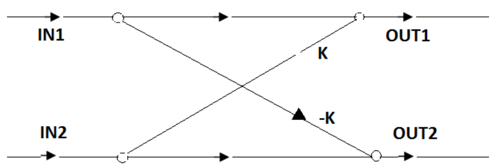

Figure 1: Lattice stage

is clear that two computations are performed at once so basically a crisscross is involved here.

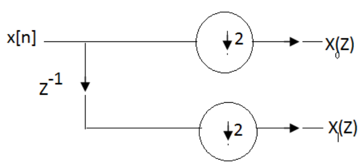

Figure 2: Operations performed in a lattice stage

Let us now relate IN1 and IN2 with OUT1 and OUT2 and we are going to do it by thinking of a 2×2 matrix known as polyphase matrix.
Let us introduce the idea of **polyphase matrices**.
At the beginning of every lattice stage, operation as shown in Figure 2 is performed. Now we seek a relation between $X(Z)$, $X_0(Z)$, $X_1(Z)$.
Graphically

$$x_{-n} \cdots\cdots\cdots\cdots x_{-2}\, x_{-1}\, x_0\, x_1 \cdots\cdots$$

The subscripts here represents the value of n.
On the X_0 branch,
$x_0[n]$ we have ....x_{-4} x_{-2} x_0 x_2 x_4........
On the X_1 branch,
$x_1[n]$ we have ....x_{-3} x_{-1} x_1 x_3 x_5........
$x[n]$ is obtained by interleaving the sequence on x_0 branch and than on x_1 branch and continue this further.

$$x_0[n] = x[2n] \qquad \forall n \in \mathbb{Z}$$
$$x_1[n] = x[2n+1] \qquad \forall n \in \mathbb{Z}$$

It is easy to see

$$X(z) = \sum_{n=-\infty}^{n=\infty} x[n]z^{-n}$$

can be decomposed into two summations.

$$= \sum_{n=-\infty}^{n=\infty} x[2n]z^{-2n} + \sum_{n=-\infty}^{n=\infty} x[2n+1]z^{-(2n+1)}$$
$$= \sum_{n=-\infty}^{n=\infty} x_0[n](z^2)^{-n} + \sum_{n=-\infty}^{n=\infty} x_1[n]z^{-1}(z^2)^{-n}$$

$x_0[n]$ and $x_1[n]$ are called the polyphase components of $x[n]$ which means switching from one phase to another one phase for one sample and other for another sample and this continues for other coming samples for constructing $x_0[n]$ and $x_1[n]$.

$$X(z) = X_0(z^2) + z^{-1}X_1(z^2)$$

So the above equation gives a relationship between Z-transform of polyphase components and Z-transform of sequence.
Lattice performs operation on this polyphase components therefore we can say that the whole of analysis and synthesis filter bank is essentially a operation on polyphase components instead of thinking as operation on sequence. It can be thought of as operation on 2×2 sequence.
So, each stage of lattice: 2×2 matrix operation on polyphase components.

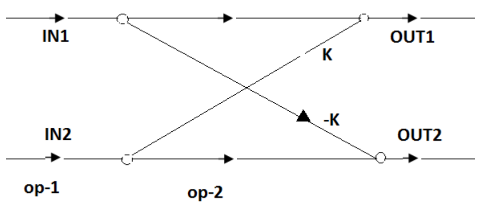

Figure 3: Operation 1

Matrix corresponding to op-1 or operation 1 as shown in Figure 3.

$$\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} IN1 \\ IN2 \end{bmatrix} = \begin{bmatrix} Int1 \\ Int2 \end{bmatrix}$$

Here Int1 and Int2 stands for intermediate stages.
Matrix corresponding to op-2 or operation 2 as shown in Figure 3

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} \begin{bmatrix} Int1 \\ Int2 \end{bmatrix} = \begin{bmatrix} OUT1 \\ OUT2 \end{bmatrix}$$

$$\begin{bmatrix} OUT1 \\ OUT2 \end{bmatrix} = \begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} IN1 \\ IN2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}$$

is known as a polyphase matrix.

$$\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}$$

can not be further simplified.
But we can think of simplifying

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix}$$

into an upper triangular matrix and a lower triangular matrix for reducing computational complexity.

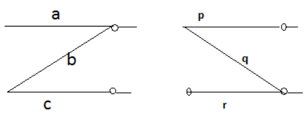

Figure 4: upper △ and lower △ computations

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & c \end{bmatrix} \begin{bmatrix} p & 0 \\ q & r \end{bmatrix}$$

Figure 4 shows the computation involve in upper △ and lower △.
On solving matrix we get,

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} = \begin{bmatrix} ap + bq & br \\ cq & cr \end{bmatrix}$$

On equating both sides,

$$\begin{aligned} ap + bq &= 1 \\ br &= k \\ cq &= -k \\ cr &= 1 \end{aligned}$$

We exploit degree of freedom by choosing very simple variable.
$p = 1$, $r = 1$ whereupon

$$\begin{aligned} a + bq &= 1 \\ cq &= -k \\ b &= k \\ c &= 1 \\ q &= -k \\ a &= 1 + k^2 \end{aligned}$$

So finally we get following matrix. Structure is represented in Figure 5.

$$\begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} = \begin{bmatrix} 1+k^2 & k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -k & \end{bmatrix}$$

On redrawing and including delay factor as shown in the Figure 6.

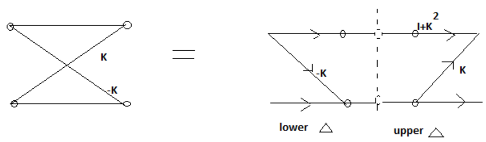

Figure 5: Resulting structure after solving matrices

The idea of lifting is to lift from no transform to a meaningful transform. In case of up

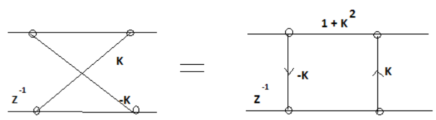

Figure 6: lifting stage

samplers and down samplers as shown in Figure 6.

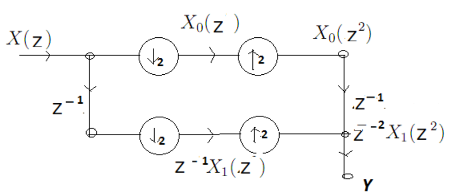

Figure 7: Lazy wavelet transform

This is called a lazy wavelet transform. It does nothing at all if we were having no lattice stages we would have structure as shown in Figure 7. So from a structure which does nothing at all we are building stage by stage a structure which has meaningful frequency response. Lifting is used because its lift the inefficient wavelet transform to a transform which does great deal in time and frequency.