Module 9 : Numerical Relaying II : DSP Perspective

Lecture 36 : Fast Fourier Transform

Objectives

In this lecture,

- We will introduce Fast Fourier Transform (FFT).
- We will show equivalence between FFT and Sequence transformation.

36.1 Fast Fourier Transform

We have seen that N- point DFT is given by the following expression

$$X(m) = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi nm}{N}} \quad (1)$$

Let $a = e^{\frac{j2\pi}{N}}$, the N^{th} root of unit. Then the following relationships can be easily derived.

1. $1 + a + a^2 + - - - - - + a^{N-1} = 0$ (2)

   Proof: using Geometric progression series formula

   $$x + xr + xr^2 + - - - - + xr^{N-1} = \frac{x(1 - r^N)}{(1 - r)}$$

   We get $1 + a + a^2 + - - - - + a^{N-1} = \frac{1(1 - a^N)}{(1 - a)} = \frac{(1 - 1)}{(1 - a)} = 0$ because $a^N = 1$ (3)

2. From $a^m a^{N-m} = 1$ and $a^m (a^m)^* = 1$, we get

   $$(a^m)^* = a^{N-m} \quad (4)$$

   Hence $a^* = a^{N-1}$, $(a^2)^* = a^{N-2}$ etc.

3. From the fact $a^{-1} a = 1$ and $a^* a = 1$, we have (5)

   $$a^{-1} = a^* = a^{N-1} \text{ and } a^{-2} = (a^*)^2 = a^{N-2} \text{ etc.} \quad (6)$$

Now using the a operator, DFT for $m = 0, - - - -, N - 1$ can be written as follows.

$$X(0) = x_0 + x_1 + ----- + x_{N-1}$$
$$X(1) = x_0 + x_1 a^{-1} + --- + x_{N-1}(a^{-1})^{N-1}$$
$$-------------------------$$
$$X(N-1) = x_0 + x_1 a^{-(N-1)} + ----- + x_{N-1}(a^{-1})^{(N-1)^2}$$

Arranging it in a matrix format, we get the following

$$\begin{bmatrix} X(0) \\ X(1) \\ | \\ | \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & - & - & 1 \\ 1 & a^{-1} & & & (a^{-1})^{N-1} \\ & & & & \\ & & & & \\ 1 & (a^{-1})^{N-1} & - & - & (a^{-1})^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ | \\ | \\ x_{N-1} \end{bmatrix} \qquad (7)$$

(or) stated more compactly

$$[X] = [P][x]$$

Where $P(i, j) = (a^{-1})^{(i-1)(j-1)}$

The i^{th} row of the matrix P indexes the i^{th} frequency component while the j^{th} column of the $P-$ matrix indexes the j^{th} sample. The matrix $P-$ enjoys a very special property viz. it's columns or rows are orthogonal to each other. If p_i and p_j denote the i^{th} and j^{th} column of matrix P, then, it is easy to verify that

$$p_i^H p_j = 0 \quad (i \neq j)$$

$$p_i^H p_i = N$$

Where H - indicates Hermitian operator defined as
$$p_i^H = (p_i^T)^* = (p_i^*)^T$$

i.e. each column is first transposed to a row and every element is then replaced by its complex conjugate.

For a real number, the complex conjugate is identical to the original number. Hence on real-valued vectors, Hermitian and transpose operators are one and the same. However, for complex valued vectors the two differ.
It is now easy to verify that

$$P^{-1} = \frac{1}{N} P^H \text{ and } P^H = \begin{bmatrix} 1 & 1 & 1 & - & 1 \\ 1 & a & a^2 & - & a^{N-1} \\ & & & & \\ & & & & \\ 1 & a^{(N-1)} & a^{2(N-1)} & - & a^{(N-1)^2} \end{bmatrix}$$

Thus,

$$[X] = [P][x] \quad (8)$$

$$[x] = \frac{1}{N}[P^H][X] \quad (9)$$

The invertibility of P, in essence captures the transform property of DFT.

In relaying, typically we are not interested in deriving all possible frequency components. Our interest is primarily in extracting the fundamental and sometimes 2^{nd} and 5^{th} harmonic (differential protection). Since, matrix-vector product involves N^2 multiplications, and similar number of additions, we say that extracting all possible frequency components by (1) would involve $O(N^2)$ (read as order N^2) effort.

This effort is considered to be significantly high for real-time computing. However, with some ingenuity, researchers have shown that the task can be achieved in approximately $\frac{N}{2}\log_2 N$ computations. This fast approach to computing all possible frequency transforms in discrete domain is called Fast Fourier Transform (FFT). For example, with $N = 8$ brute force implementation of (1) requires 64 complex multiplications which can be reduced to 12 multiplications with FFT.

As we would not have much use for FFT in this course, we will not pursue this topic any further. Rather, we now establish an equivalence between two very well known transforms viz. multiple DFT (or) FFT and sequence component transformation. [for the remaining text through out we will refer equation (1) as FFT].

36.2 Equivalence of FFT and N-phase sequence component transformation:

We will first review N-phase sequence transformation. Consider a N-phase (balanced or unbalanced) system ($N \geq 3$). Let the phasors in phase domain e.g. (V_a, V_b, V_c for 3-phase system) be represented by $\bar{x}_1, \bar{x}_2, \ldots. \bar{x}_N$. Note that $\bar{x}_1, \bar{x}_2, \ldots. \bar{x}_N$ are complex numbers. Then, these phasors can be expressed as a linear combination of N-set of balanced N-phase systems as follows.

0-sequence component

$$X_0 = \frac{\bar{x}_1 + \bar{x}_2 + --- + \bar{x}_N}{N}$$

There are n- such phasors in the 0-sequence system each of equal magnitude and angle.

1-sequence component

$$X_1 = \frac{\bar{x}_1 + \bar{x}_2 a + \bar{x}_3 a^2 + --- + \bar{x}_N a^{N-1}}{N} \quad (10)$$

There are n- such phasors in 1-sequence system. If $X_{1,1}$ is taken as reference, then for a balanced system $X_{1,2}$ is equal in magnitude to $X_{1,1}$ but phase shifted by angle $-\frac{2\pi}{N}$ i.e. $X_{1,2} = X_{1,1}a^{N-1}$

Similarly, 2-sequence component

$$X_2 = \frac{\bar{x}_1 + \bar{x}_2 a^2 + \bar{x}_3 a^4 + --- + \bar{x}_N (a^2)^{N-1}}{N} \quad (11)$$

In this system b-phase lags a-phase by $-\frac{4\pi}{N}$. Proceeding in the similar manner we get,

$$X_{N-1} = \frac{\bar{x}_1 + \bar{x}_2 (a)^{N-1} + \bar{x}_3 (a^2)^{N-1} + - - - + \bar{x}_N a^{(N-1)^2}}{N} \tag{12}$$

Equivalence of FFT and N-phase sequence component transformation: (contd..)

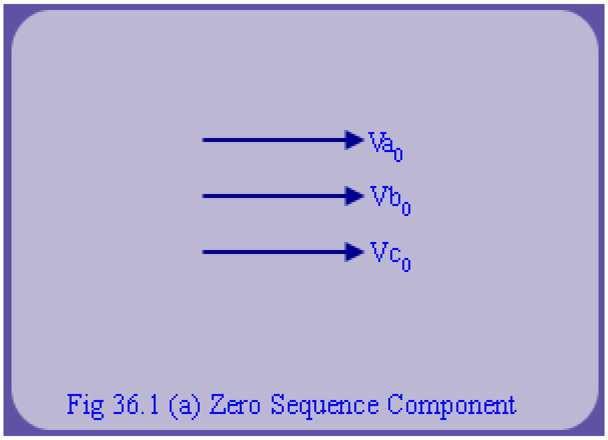

Fig 36.1 (a) Zero Sequence Component

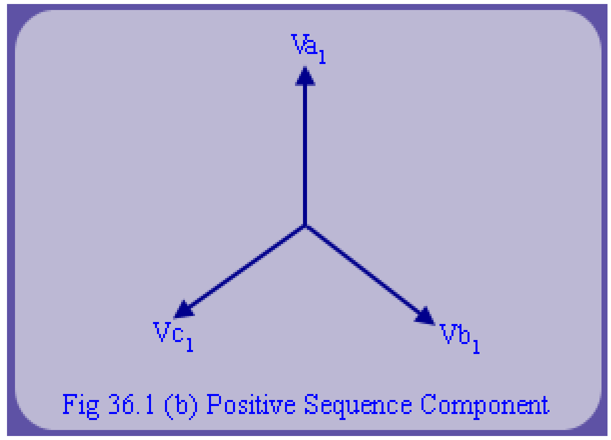

Fig 36.1 (b) Positive Sequence Component

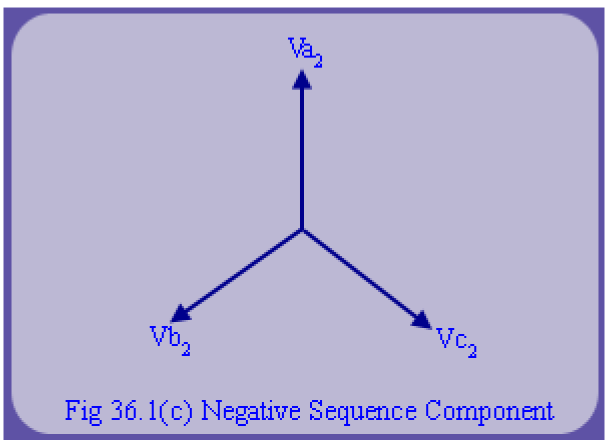

Fig 36.1(c) Negative Sequence Component

Fig 36.1 (a, b and c) visualize the system for a 3-phase system. Expressing these equations in matrix format we obtain the following equations.

$$\begin{bmatrix} X_0 \\ X_1 \\ | \\ | \\ X_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & - & 1 \\ 1 & a & a^2 & - & a^{N-1} \\ 1 & a^2 & a^4 & - & a^{2(N-1)} \\ | & | & & & \\ 1 & a^{(N-1)} & a^{2(N-1)} & & a^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ | \\ | \\ \bar{x}_N \end{bmatrix} \tag{13}$$

Equivalence of FFT and N-phase sequence component transformation: (contd..)

Thus, writing compactly

$$[X] = [P^{-1}][\bar{x}]$$

or $[\bar{x}] = [P][X]$ (14)

Where $\overline{X} = \left[X_0, X_1, \ldots\ldots, X_{N-1}\right]^T$

$\overline{x} = \left[x_1, x_2, \ldots\ldots, x_N\right]^T$

Thus, from (8) and (14) we conclude that FFT and sequence transformation (defined from sequence domain to the phase domain) involve the same transformation matrix P. Hence, the two transforms are mathematically equivalent. In particular for N = 3. With a- phase as reference phasor, we see the following equivalence relationships.

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & a^2 & a \\ \mathbf{1} & a & a^2 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix}$$

and from DFT, we get

$$\begin{bmatrix} V(0) \\ V(1) \\ V(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

This mathematical equivalence brings out an important concept viz. transformations and decompositions done via orthogonal matrices can have multiple interpretations. However, there is one important difference between the two transformations. The samples $x_0, \ldots\ldots, x_{N-1}$ in signal processing are usually real numbers while corresponding phasors in sequence analysis are complex numbers. Thus, while there is a redundancy in information in FFT domain which leads to DFT symmetry property, there is no such redundancy in sequence domain. Hence, in sequence domain we do not come across such a property.

The mathematical equivalence of two should put the reader at ease with both these transformations, irrespective of which one he came across first.

Recap

In this lecture, we have

- Shown equivalence of generic FFT and N-phase sequence transformation.

- Illustrated differences between the properties of the 2 - transforms , because first transform (FFT) is used to

  convert n-dimensional real vector (R^n) to n-dimensional complex vector (C^n) while the N-phase sequence transformation maps a vector to a C^n vector.