4. The Discrete Fourier Transform and Fast Fourier Transform

- *Reference*: Sections 8.0-8.7 of Text

  Note that the text took a different point of view towards the derivation and the interpretation of the discrete Fourier Transform (DFT). Our derivation is more "direct".

- In many situations, we need to determine numerically the frequency response of an analog system or the spectrum of an analog signal. Let $x(t)$ be the signal under consideration. Then the procedure usually adopted is

  1. Select a suitable sampling fequency $f_s = 1/T$ according to the Nyquist sampling criterion.

  2. Sample $x(t)$ at time $t = nT$ to obtain the discrete time (DT) signal $x[n]$.

  3. If the discrete time signal is infinitely long, truncate it to a suitable length, say N samples. Call the resultant signal $\hat{x}[n]$.

  4. Compute the Fourier transform (assuming a casual signal)

  $$\hat{X}\left(e^{j\mathbf{w}}\right) = \sum_{n=0}^{N-1} \hat{x}[n]e^{-j\mathbf{w}n} \tag{1}$$

  at the desired frequencies. Note that if there is no need to truncate the signal, then

  $$\hat{X}\left(e^{j\mathbf{w}}\right) = X\left(e^{j\mathbf{w}}\right) = \frac{1}{T} X_a\left(j\frac{\mathbf{w}}{T}\right)$$

where $X_a(j\Omega)$ is the Fourier transform of the analog signal $x(t)$. In general,

$$\hat{X}\left(e^{jw}\right) \approx \frac{1}{T} X_a\left(j\frac{w}{T}\right)$$

- If the signal whose spectrum we want to deterime is a discrete time signal, then Steps 1 & 2 in the above procedure is no longer needed.

- This chapter is concerned with the efficient computation of Eqn (1) at discrete frequencies

$$w = 2pk/N; \qquad k = 0,1,...,N-1$$

For convience, we will drop the sign ^ associated with $\hat{X}\left(e^{jw}\right)$ and simply use $X\left(e^{jw}\right)$ instead.

4.1 The Discrete Fourier Transfomr (DFT)

- The DFT of a finite duration (and casual) signal is defined as

$$X[k] = X\left(e^{jw}\right)\Big|_{w=2pk/N} = \sum_{n=0}^{N-1} x[n]e^{-j(2pk/N)n}$$

$$= \sum_{n=0}^{N-1} x[n]W_N^{kn}; \qquad\qquad k = 0,1,...,N-1$$

where

$$W_N = e^{-j2p/N}$$

- In addition to being a sample of the FT at $w = 2pk/N$, the DFT coefficient $X[k]$ also represents a sample of the ZT on the unit circle.

- The inverse discrete Fourier transform (IDFT) is

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}; \qquad\qquad n = 0,1,...,N-1$$

Proof:

$$\frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} = \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} x[m] W_N^{km} \right) W_N^{-kn}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} x[m] \left(\sum_{k=0}^{N-1} W_N^{k(m-n)} \right)$$

But

$$\sum_{k=0}^{N-1} W_N^{k(m-n)} = \frac{1 - W_N^{(m-n)N}}{1 - W_N^{(m-n)}} = \frac{1 - e^{-j(2p/N)(m-n)N}}{1 - e^{-j(2p/N)(m-n)}}$$

$$= \begin{cases} N & m = n \\ 0 & m \neq n \end{cases}$$

So

$$\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-kn}=x[n]$$

- The DFT and the IDFT pair can also be represented in matrix form as

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{(N-1)2} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

or

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

where

$$\mathbf{x} = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{(N-1)2} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

Note that

$$\mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}^{\dagger}$$

where $(\cdot)^{\dagger}$ denotes the Hermitian (or conjugate) transpose of a matrix.

- We can deduce from the matrix representation of the DFT that its computational complexity is in the order of $O(N^2)$.

- The Fast Fourier Transform (FFT) is an efficient algorithm for the computation of the DFT. It only has a complexity of $O(N \log N)$.

- From the DFT coefficients, we can compute the FT at any frequency. Specifically

$$X\left(e^{j\mathbf{w}}\right) = \sum_{n=0}^{N-1} x[n] e^{-j\mathbf{w}n}$$

$$= \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \right] e^{-j\mathbf{w}n}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left[\sum_{n=0}^{N-1} e^{-j(\mathbf{w}-2\mathbf{p}k/N)n} \right]$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \frac{1 - e^{-j(\mathbf{w}-2\mathbf{p}k/N)N}}{1 - e^{-j(\mathbf{w}-2\mathbf{p}k/N)}}$$

$$= \frac{1 - e^{-j\mathbf{w}N}}{N} \sum_{k=0}^{N-1} \frac{X[k]}{1 - e^{-j\mathbf{w}} e^{j2\mathbf{p}k/N}}$$

- Since the FT is the ZT evaluated on the unit circle, this means we can also obtain the ZT from the DFT coefficients by replacing the term $e^{j\mathbf{w}}$ in the above equation by z. In other word

$$X(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X[k]}{1 - z^{-1} e^{j2\pi k/N}}$$

4.1.1 Convolution of Sequences

- Let $x_1[n]$ and $x_2[n]$ be two DT signals of duration N samples. We want to obtain their convolution:

$$y[n] = x_1[n] \otimes x_2[n]$$

- Direct convolution leads to a complexity in the order of $O(N^2)$. With the availability of the FFT, it is possible to perform the same task with a complexity of only $O(N \log N)$.

  The idea is to first multiply the DFT coefficients of the two signals together and then take an IDFT of the product.

  This ideal, although simple, has to be exercised carefully.

- Let $X_1(e^{jw})$, $X_2(e^{jw})$, and $Y(e^{jw})$ be the FTs of $x_1[n]$, $x_2[n]$, and $y[n]$ respectively. Then

$$Y(e^{jw}) = X_1(e^{jw}) X_2(e^{jw})$$

- From the definition of the DFT, we know

$$\hat{X}_1[k] = \sum_{n=0}^{N-1} x_1[n] W_N^{kn} = X_1\left(e^{jw}\right)\Big|_{w=2\boldsymbol{p}\,k/N},$$

$$\hat{X}_2[k] = \sum_{n=0}^{N-1} x_2[n] W_N^{kn} = X_2\left(e^{jw}\right)\Big|_{w=2\boldsymbol{p}\,k/N}$$

So we can deduce that

$$\hat{Y}[k] = Y\left(e^{jw}\right)\Big|_{w=2\boldsymbol{p}\,k/N}$$
$$= \hat{X}_1[k]\,\hat{X}_2[k]$$

- While $\hat{Y}[k]$, $k = 0,1,...,N-1$, are samples of $Y\left(e^{jw}\right)$, we will not be able to compute $y[n]$ from them through an IDFT operation. To see this, let

$$\hat{y}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \hat{Y}[k\,]W_N^{-kn}$$
$$= \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{m=-\infty}^{\infty} y[m]e^{-j2\boldsymbol{p}km/N}\right)e^{j2\boldsymbol{p}kn/N}$$
$$= \frac{1}{N}\sum_{m=-\infty}^{\infty} y[m]\left(\sum_{k=0}^{N-1} e^{j2\boldsymbol{p}k(n-m)/N}\right)$$
$$= \frac{1}{N}\sum_{m=-\infty}^{\infty} y[m]\frac{1-e^{j2\boldsymbol{p}k(n-m)}}{1-e^{j2\boldsymbol{p}k(n-m)/N}}$$

Since

$$\frac{1 - e^{j2\pi k(n-m)}}{1 - e^{j2\pi k(n-m)/N}} = \begin{cases} N & m = n + rN \\ 0 & \text{otherwise} \end{cases},$$
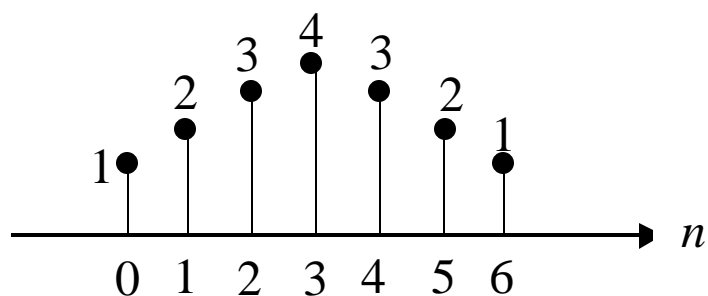
the above becomes

$$\hat{y}[n] = \sum_{r=-\infty}^{\infty} y[n + rN]$$

In other word, $\hat{y}[n]$ is an aliased version of $y[n]$.

- **Example**: Let

$$x_1[n] = x_2[n] = \begin{cases} 1 & 0 \le n \le 3 \\ 0 & \text{otherwise} \end{cases}.$$

Then $y[n]$ is a triangular signal

and $\hat{y}[n] = 4$, $n = 0,1,2,3$.

- The cause for aliasing in $\hat{y}[n]$ is the lack of sufficient DFT coefficients in its construction. Note that $y[n]$ is of duration $2N-1$ samples so we need as many DFT coefficients to capture all the information about the signal.

- To eliminate aliasing, we first treat $x_1[n]$ and $x_2[n]$ as signals with a duration of

$$M = 2N$$

samples, where $x_1[n] = x_2[n] = 0$ when $N \leq n \leq 2N-1$. Note that in general, M can be any number as long as it is not smaller than the duration of $y[n]$.

We then take the M-point DFTs of $x_1[n]$ and $x_2[n]$ to obtain

$$X_1[k] = \sum_{n=0}^{M-1} x_1[n] W_M^{nk} = X_1\left(e^{jw}\right)\Big|_{w=pk/N},$$

$$X_2[k] = \sum_{n=0}^{M-1} x_2[n] W_M^{nk} = X_2\left(e^{jw}\right)\Big|_{w=pk/N},$$

After this, we form the product

$$Y[k] = X_1[k]X_2[k] = Y\left(e^{jw}\right)\Big|_{w=pk/N}, \qquad k = 0,1,...,M-1$$

Finally, we calculate the IDFT

$$\frac{1}{N}\sum_{k=0}^{M-1}Y[k]W_M^{-kn} = \sum_{r=-\infty}^{\infty}y[n+rM] = \sum_{r=-\infty}^{\infty}y[n+2rN]$$

Since the duration of $y[n]$ is $2N-1$, which is less than M, the term $y[n+2rN]$ in the above equation equals to 0 for any $r \neq 0$. Consequently

$$\frac{1}{N}\sum_{k=0}^{M-1}Y[k]W_M^{-kn} = y[n]$$

and there is no more aliasing.

4.1.2. Cicular Convolution and Aliasing

- We show in Section 4.1.1 that if we take the IDFT of the product coefficients $\hat{Y}[k] = \hat{X}_1[k]\hat{X}_2[k]$, $k = 0,1,...,N-1$, we obtain

$$\hat{y}[n] = \sum_{r=-\infty}^{\infty}y[n+rN] \qquad (2)$$

which is an aliased version of $y[n]$.

- We will arrive at the same result through circular convolution of periodic signals constructed from $x_1[n]$ and $x_2[n]$.

- Let

$$\tilde{x}_1[n] = x_1\left[(n)_N\right]$$

and

$$\tilde{x}_2[n] = x_2\left[(n)_N\right],$$

where $(n)_N$ is the modulo-N value of the integer n.

Because of the property of the modulo-N operation, the signals $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$ are periodic with a period of N. Specifically,

$$\tilde{x}_1[m + rN] = x_1[m], \qquad m = 0,1,..., N-1$$

and

$$\tilde{x}_2[m + rN] = x_2[m], \qquad m = 0,1,..., N-1$$

for any integer r.

- Conversely, the signals $x_1[n]$ and $x_2[n]$ can be obtained from $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$ according to

$$x_1[n] = \tilde{x}_1[n]R_N[n]$$

and

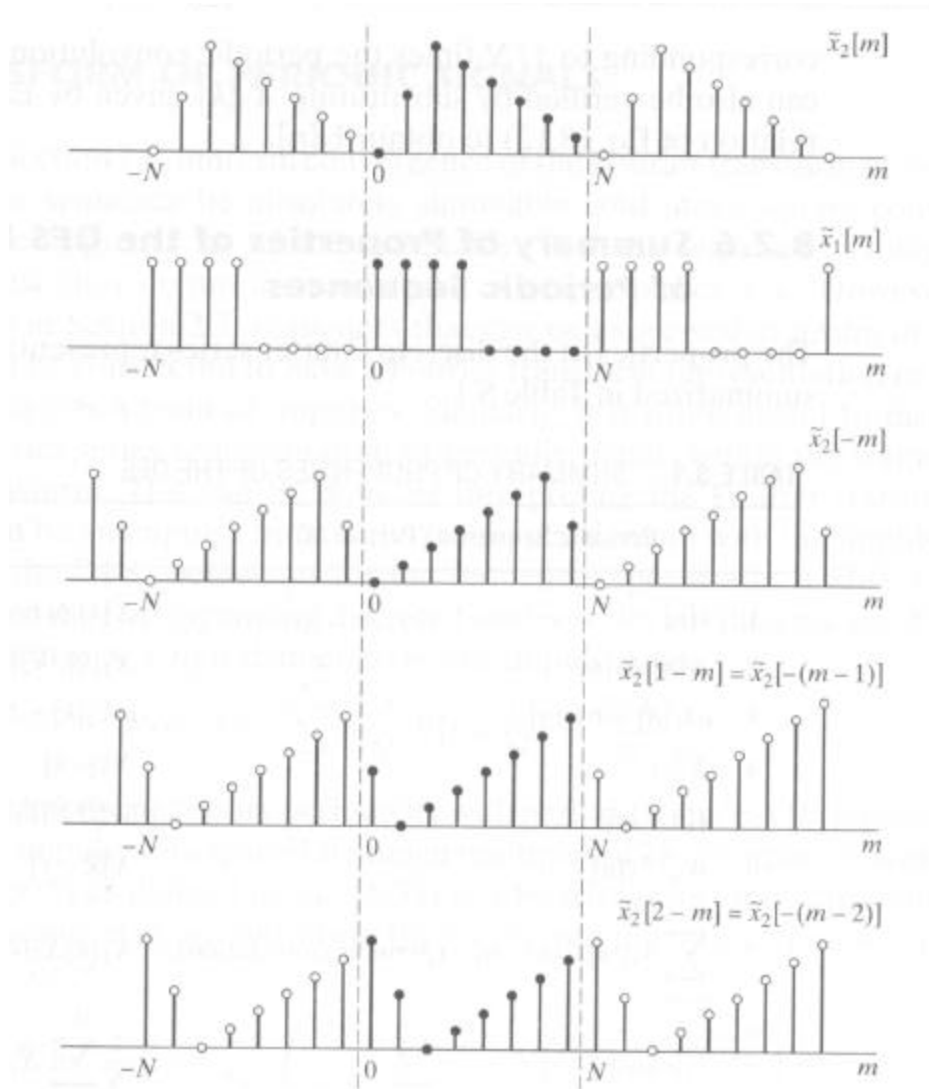$$x_2[n] = \tilde{x}_2[n]R_N[n],$$

where

$$R_N[n] = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

is a rectangular windowing function.

- The circular or periodic convolution of $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$ is defined as

$$\tilde{x}_3[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m]$$

Clearly, $\tilde{x}_3[n]$ is a periodic signal with a period of N.

- We want to show that the signal $\hat{y}[n]$ in Eqn (2) is simply one period of $\tilde{x}_3[n]$, i.e.

$$\hat{y}[n] = \tilde{x}_3[n]R_N[n]$$

Proof:

- Since the signal $\tilde{x}_3[n]$ is periodic, we resctrict our attention to time indices n between 0 and N-1.

- Since $0 \leq m \leq N-1$, the term $\tilde{x}_1[m]$ in the circular convolution is simply $x_1[m]$.

- The index $n-m$ of the term $\tilde{x}_2[n-m]$ can be as small as $1-N$ or as large as $N-1$. When $n-m \geq 0$,

$$\tilde{x}_2[n-m] = x_2[n-m]; \qquad n-m \geq 0$$

On the other hand when $n-m$ is negative, then

$$\tilde{x}_2[n-m] = x_2[N+n-m]; \qquad n-m < 0$$

- The circular convolution can now be rewritten as:

$$\tilde{x}_3[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m]$$

$$= \sum_{m=0}^{n} \tilde{x}_1[m]\tilde{x}_2[n-m] + \sum_{m=n+1}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m]$$

$$= \sum_{m=0}^{n} x_1[m]x_2[n-m] + \sum_{m=n+1}^{N-1} x_1[m]x_2[N+n-m]$$

$$= \sum_{m=0}^{n}\left(\frac{1}{N}\sum_{k=0}^{N-1}\hat{X}_1[k]W_N^{-km}\right)\left(\frac{1}{N}\sum_{r=0}^{N-1}\hat{X}_2[r]W_N^{-r(n-m)}\right)$$

$$+ \sum_{m=n+1}^{N-1}\left(\frac{1}{N}\sum_{k=0}^{N-1}\hat{X}_1[k]W_N^{-km}\right)\left(\frac{1}{N}\sum_{r=0}^{N-1}\hat{X}_2[r]W_N^{-r(N+n-m)}\right)$$

$$= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{r=0}^{N-1}\hat{X}_1[k]\hat{X}_2[r]W_N^{-nr}\left(\sum_{m=0}^{n}W_N^{(r-k)m}\right)$$

$$+ \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{r=0}^{N-1}\hat{X}_1[k]\hat{X}_2[r]W_N^{-(n+N)r}\left(\sum_{m=n+1}^{N-1}W_N^{(r-k)m}\right)$$

$$= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{r=0}^{N-1}\hat{X}_1[k]\hat{X}_2[r]W_N^{-nr}\left(\sum_{m=0}^{N-1}W_N^{(r-k)m}\right)$$

But

$$\sum_{m=0}^{N-1}W_N^{(r-k)m} = \begin{cases} N & r=k \\ 0 & \text{otherwise} \end{cases}$$

So

$$\tilde{x}_3[n] = \frac{1}{N}\sum_{k=0}^{N-1}\hat{X}_1[k]\hat{X}_2[k]W_N^{-nk}$$

$$= \text{IDFT}\left\{\hat{X}_1[k]\hat{X}_2[k]\right\}$$

$$= \hat{y}[n]$$

for $0 \le n \le N-1$

- We will discuss more about circular convolution in the section on OFDM modulation.

4.1.2 The DFT of an infinite-duration signal

- The DFT is defined only for a finite duration signal. As mentioned earlier, when the signal is of infinite duration, we must first truncate the signal to a suitable length before taking DFT. Truncation is equivalent to multiplying the infinitely long signal by the rectangular window function

$$w[n] = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

- Let $x[n]$ and $y[n] = x[n]w[n]$ be the signals before and after truncation. As shown in Chapter 2, the FT of $y[n]$ is

$$Y\left(e^{jw}\right) = \frac{1}{2\mathbf{p}} \int_{-\mathbf{p}}^{\mathbf{p}} X\left(e^{j\mathbf{q}}\right) W\left(e^{j(w-\mathbf{q})}\right) d\mathbf{q} \ ,$$

where $X\left(e^{jw}\right)$ and $W\left(e^{jw}\right)$ are respectively the FTs of the signal $x[n]$ and the windowing function $w[n]$.

- The DFT coefficients of the signal $y[n]$ are

$$Y[k] = \sum_{n=0}^{N-1} y[n] W_N^{kn}$$

$$= \sum_{n=0}^{N-1} x[n] W_N^{kn}$$

$$= Y\left(e^{jw}\right)\Big|_{w=2\mathbf{p}k/N}; \qquad k = 0,1,...,N-1$$

- Since $Y(e^{jw}) \neq X(e^{jw})$, so the $Y[k]$s are not exactly the samples of $X(e^{jw})$ at $w = 2\pi k / N$.

  The spectral distortion can be reduced by using a longer truncation length and/or a different windowing function.

- Some commonly used windowing functions are

  1. Hamming,
  2. Hanning,
  3. Kaiser, and
  4. Blackman

  We will talk more about them in the Chapter on FIR filter design.

- **Example**: Let the original signal be

$$x[n] = a^n u[n].$$

  where the parameter a is real, positive, and less than 1. The FT of this signal is

$$X(e^{jw}) = \frac{1}{1 - ae^{-jw}}$$

  Now, if we truncate this signal to N samples, we obtain the signal

$$y[n] = \begin{cases} x[n] & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

The DFT coefficients of this signal is

$$
\begin{aligned}
Y[k] &= \sum_{n=0}^{N-1} y[n]W_N^{kn} = \sum_{n=0}^{N-1} a^n W_N^{kn} = \sum_{n=0}^{N-1}\left(ae^{-j2\pi k/N}\right)^n \\
&= \frac{1-\left(ae^{-j2\pi k/N}\right)^N}{1-ae^{-j2\pi k/N}} \\
&= \frac{1-a^N}{1-ae^{-j2\pi k/N}} \\
&= \left(1-a^N\right)X\left(e^{j\omega}\right)\Big|_{\omega=2\pi k/N}
\end{aligned}
$$

4.1.4 The DFT of a 2-sided signal

- In the definition of the DFT, the signal is implicitly assumed casual. In this case,

$$
X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} = X\left(e^{j\omega}\right)\Big|_{\omega=2\pi k/N}\ ; \qquad k=0,1,...,N-1
$$

What happens when the signal is not casual, like the discrete-time version of the SQRC pulse used in our project?

- Let $x[n]$ be a non-casual signal time-limited to the range $\left[-\frac{N}{2}, \frac{N}{2}-1\right]$, where the duration N is assumed to be an even number.

One possible definition of the DFT of this signal is

$$\hat{X}[k] = \sum_{n=0}^{N-1} x\left[n - \tfrac{N}{2}\right] W_N^{kn}$$

$$= \sum_{n=0}^{N-1} x\left[n - \tfrac{N}{2}\right] e^{-j2\boldsymbol{p}kn/N}$$

$$= \sum_{n=0}^{N-1} x\left[n - \tfrac{N}{2}\right] e^{-j2\boldsymbol{p}k(n-N/2)/N}\, e^{-j\boldsymbol{p}k}$$

$$= \sum_{m=-N/2}^{N/2-1} x[m] e^{-j2\boldsymbol{p}km/N}\, e^{-j\boldsymbol{p}k} \qquad .$$

$$= e^{-j\boldsymbol{p}k} X\left(e^{j\boldsymbol{w}}\right)\Big|_{\boldsymbol{w}=2\boldsymbol{p}k/N}$$

However, this is not exactly $X\left(e^{j\boldsymbol{w}}\right)\big|_{\boldsymbol{w}=2\boldsymbol{p}k/N}$.

- In order that the k-th DFT coefficient equals $X\left(e^{j\boldsymbol{w}}\right)$ at $\boldsymbol{w} = 2\boldsymbol{p}k/N$ when the signal is 2-sided, we should take instead the DFT of the signal

$$y[n] = \begin{cases} x[n] & 0 \le n \le \tfrac{N}{2} - 1 \\ x[n-N] & \tfrac{N}{2} \le n \le N-1 \end{cases}$$

**Proof**:

$$X\left(e^{j\boldsymbol{w}}\right) = \sum_{n=-N/2}^{N/2-1} x[n] e^{-j\boldsymbol{w}n}$$

$$= \sum_{n=0}^{N/2-1} x[n] e^{-j\boldsymbol{w}n} + \sum_{n=-N/2}^{-1} x[n] e^{-j\boldsymbol{w}n}$$

$$= \sum_{n=0}^{N/2-1} x[n] e^{-j\boldsymbol{w}n} + \sum_{n=N/2}^{N-1} x[n-N] e^{-j\boldsymbol{w}(n-N)}$$

4-18

Evaluating the above at $w = 2pk/N$ yields

$$X\left(e^{jw}\right)\Big|_{w=2pk/N} = \sum_{n=0}^{N/2-1} x[n]e^{-j2pkn/N} + \sum_{n=N/2}^{N-1} x[n-N]e^{-j2pk(n-N)/N}$$

$$= \sum_{n=0}^{N/2-1} x[n]W_N^{kn} + \sum_{n=N/2}^{N-1} x[n-N]W_N^{kN}$$

$$= \sum_{n=0}^{N-1} y[n]W_N^{kn}$$

$$= Y[k]$$

- In summary, for a 2-sided signal, we should first attach the negative-time portion of the signal to the positive-time portion before taking DFT. For example, if the original signal is
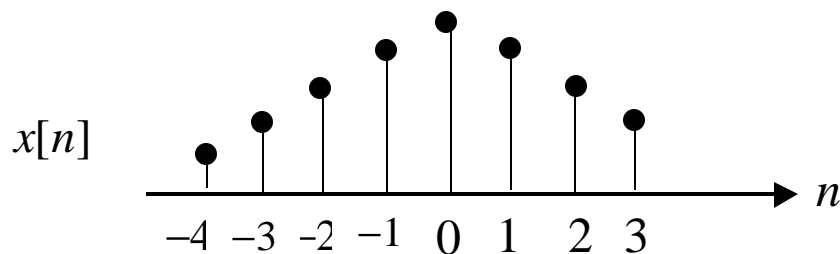
then after manupulation, we have
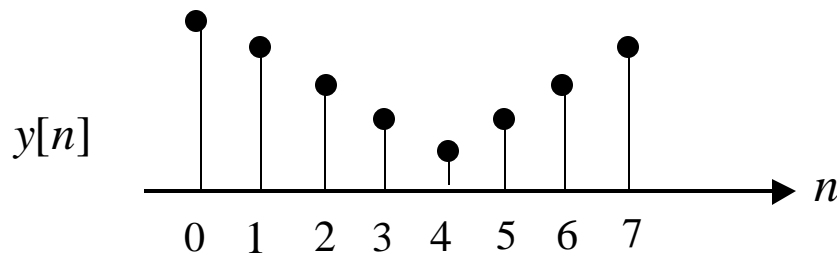
4.2 The Decimation-in-Time FFT Algorithm

- *Reference*: Sections 9.1-9.3 of Text

- As discussed in the last section, the direct method of computing the DFT coefficients has a complexity of N^2 complex multiplications and $N(N-1)$ complex additions.

  Each complex multiplication itself requires 4 real multiplications and 2 real additions.

  Each complex addition involves two real additions.

  So the total number of real multiplications is $4N^2$ and the total number of real addition is $4N^2 - 2N$.

- By decomposing the original *N*-point DFT into successively smaller DFTs (a divide-and-conquer approach), the amount of computations can be dramatically reduced. In this process, the properties of the complex exponential function

$$W_N = e^{-j2\pi/N}$$

are exploited. Specifically, we notice that

1. $W_N^{k(N-n)} = W_N^{-kn} = \left(W_N^{kn}\right)^*$ (complex conjugate symmetry)
2. $W_N^{kn} = W_N^{k(n+N)} = W_N^{n(k+N)}$ (periodicity in *n* and *k*).

The resultant algorithms are collectively known as *Fast Fourier Transform* (FFT).

- We will focus in this section on the derivation of the decimation-in-time FFT algorithm. For convenience, we assume

$$N = 2^v$$

  where v is an integer. The idea, however, can be generalized to any other composite (i.e. non-prime) value of N.

- The DFT coefficients are given by

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}; \qquad\qquad k = 0,1,...,N-1$$

  Since $N = 2^v$ is an even number, we can express the above as the sum of two terms, one involving the even-numbered $x[n]$s, and the other involving the odd-numbered $x[n]$s:

$$X[k] = \sum_{n \text{ even}} x[n] W_N^{kn} + \sum_{n \text{ odd}} x[n] W_N^{kn}$$

$$= \sum_{r=0}^{N/2-1} x[2r] W_N^{2kr} + \sum_{r=0}^{N/2-1} x[2r+1] W_N^{(2r+1)k}$$

$$= \sum_{r=0}^{N/2-1} x[2r] W_N^{2kr} + W_N^{k} \sum_{r=0}^{N/2-1} x[2r+1] W_N^{2kr}$$

- The term W_N^{2kr} can be written as

$$W_N^{2kr} = e^{-j2\mathbf{p}(2kr)/N} = e^{-j2\mathbf{p}kr/(N/2)} = W_{N/2}^{kr}$$

Consequently $X[k]$ can be rewritten as

$$X[k] = \sum_{r=0}^{N/2-1} x[2r]W_N^{2kr} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_N^{2kr}$$

$$= \sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{kr} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{kr}$$

$$= G[k] + H[k]$$

where

$$G[k] = \sum_{r=0}^{N/2-1} x[2r]W_{N/2}^{kr}; \qquad k = 0,1,...,N-1$$

and

$$H[k] = \sum_{r=0}^{N/2-1} x[2r+1]W_{N/2}^{kr}; \qquad k = 0,1,...,N-1$$

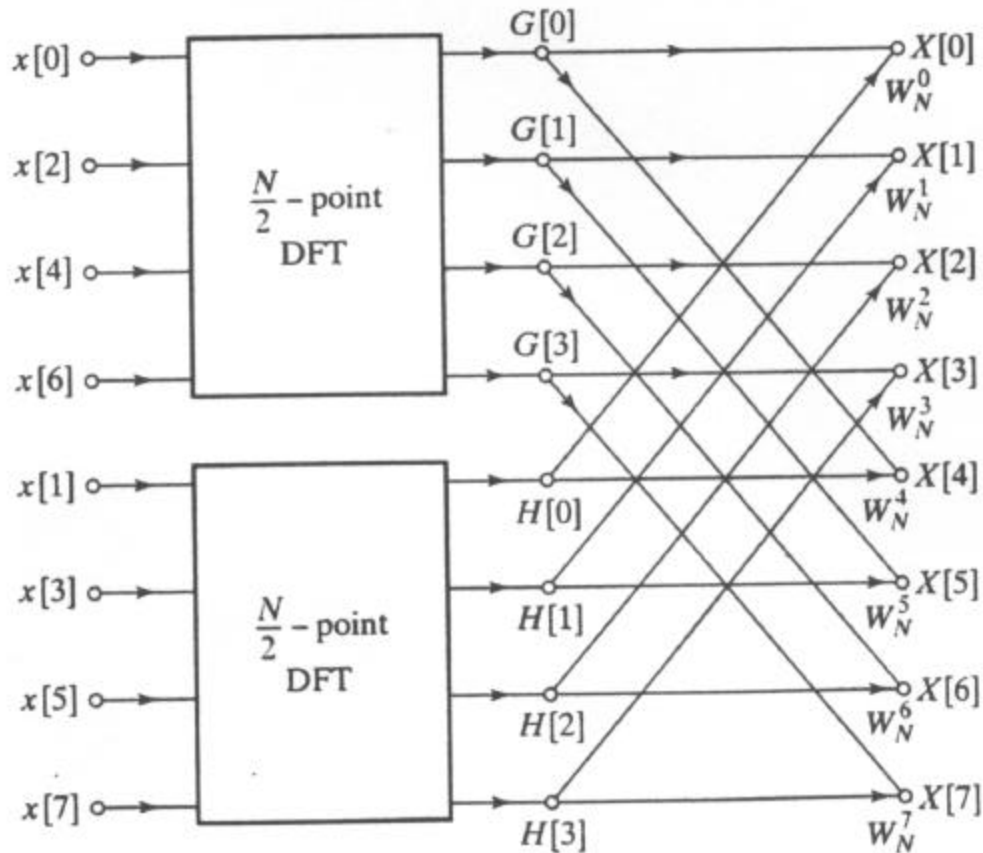- Since

$$W_{N/2}^{(m+N/2)r} = W_{N/2}^{mr},$$

this means

$$G[m+N/2] = G[m]; \qquad m = 0,1,...N/2-1$$

and

$$H[m+N/2] = H[m]; \qquad m = 0,1,...N/2-1.$$

Moreover, the $G[0], G[1],..., G[N/2-1]$ are the DFT coefficients of the subsequence $x[2n]$. Similarly, $H[0], H[1],..., H[N/2-1]$ are the DFT coefficients of the subsequence $x[2n+1]$.

- So in summary, we have at this point express the original DFT as a sum of two DFTs of half the size. Example for the case of $N=8$ is shown below

In the above diagram, the 8 DFT coefficients $X[0]$ to $X[7]$ are computed from $G[0], G[1], G[2], G[3]$ and $H[0], H[1], H[2], H[3]$ via $N/2=4$ butterfly structures. Each butterfly is of the form
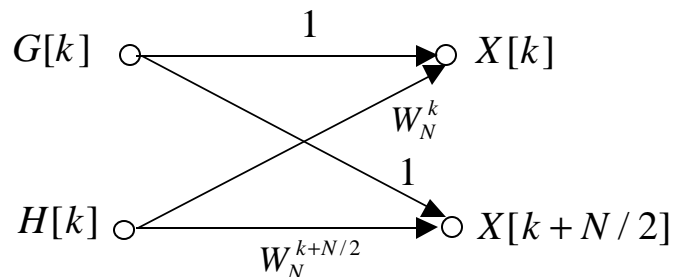
- The butterfly used in the computation of the DFT is an example of a signal flow graph (essentially a computational structure).

  In general, a signal flow graph would consist of a set of nodes or states (as represented by the circles in the butterfly structure) interconnected by a set of directed branches. The weights (as represented by the parameters 1, 1, W_N^k, and $W_N^{k+N/2}$ in the butterfly) associated with the branches are called transmittances. The transmittance from node j to node k might be denoted by t_{jk} in general.

  Associated with each node in the signal flow graph is a variable or node value. The value associated with node k might be denoted by s_k and is equal to

$$s_k = \sum_{j=1}^{L} s_j t_{jk}$$

- The butterfly shown in the previous page requires 2 complex multiplications and 2 complex additions. But since

$$W_N^{k+N/2} = -W_N^k,$$

  the number of multiplications can be reduced to 1 if the following structure is adopted instead
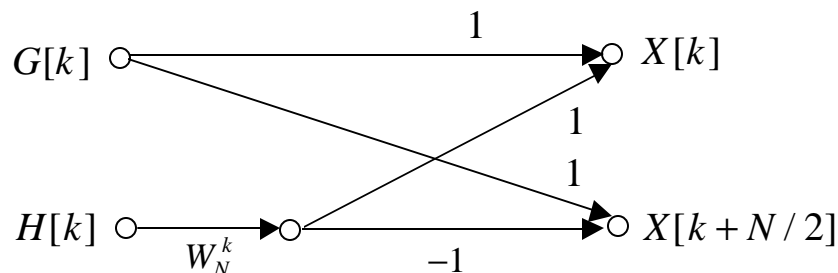
So in total, the complexity of the 4 butterflies in the above example is 4 complex multiplications (CM) and 8 complex additions (CA). In general, there are $N/2$ CMs and N CAs.

- Since $N = 2^v$, $N/2$ is also an even number. This means each of the two $N/2$-point DFT in the above example can be expressed in terms of two $N/4$-DFTs. For example, we show below the computation of $G[0], G[1], G[2], G[3]$ from $x[0], x[2], x[4], x[6]$.
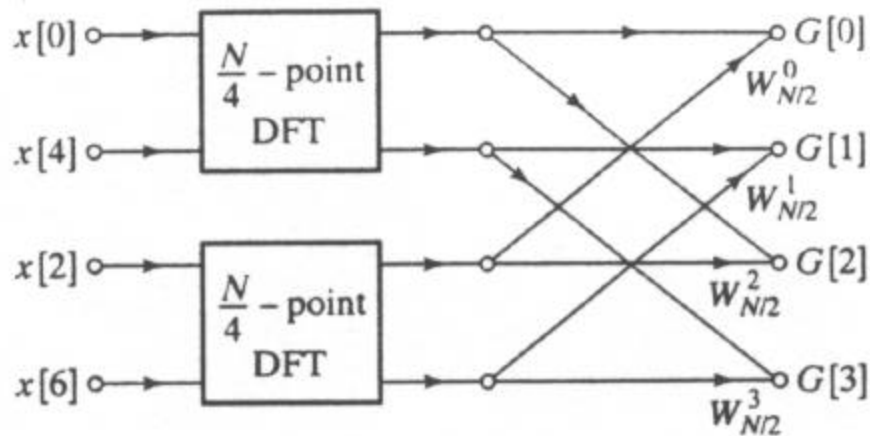
The overall computational structure now takes on the form shown in the diagram in the next page. The two $N/2$-point DFTs are now replaced by 4 butterflies.

Since the number of butterflies in the second stage (from the right) is the same as that in the first stage, the computational complexities of both stages are identical and equal to 4 CMs and 8 CAs.
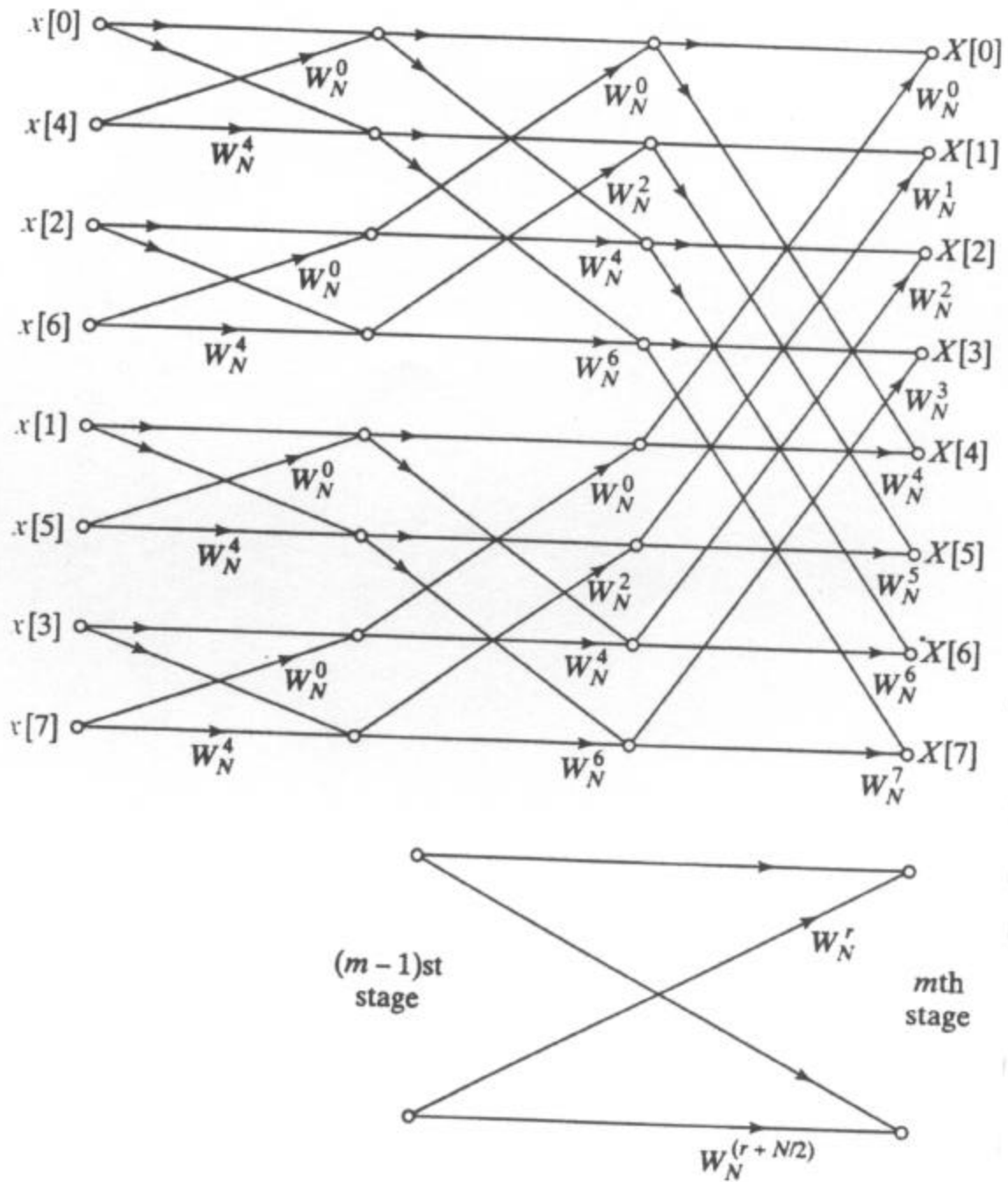
In general, (when $N = 2^v$,) there are always $N/2$ butterflies at each stage.
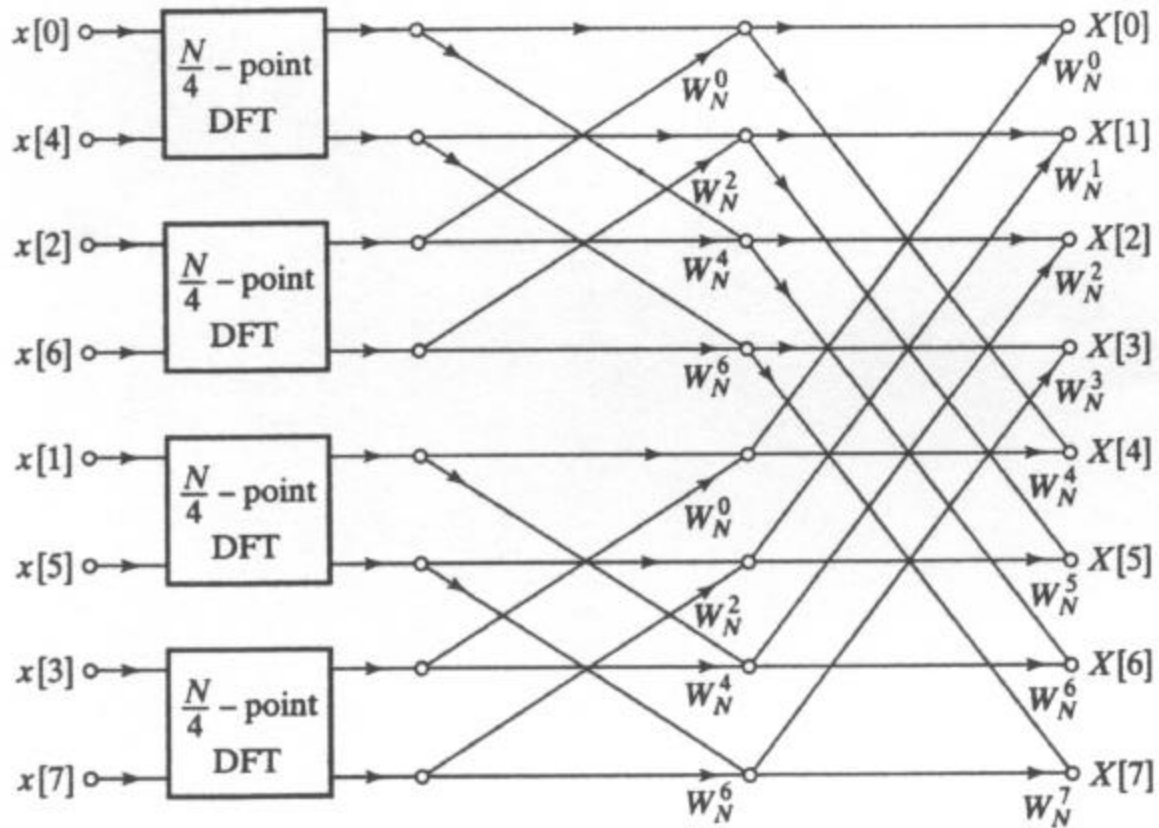
- Each of the four $N/4$-point DFTs in the above example can be computed using the following butterfly
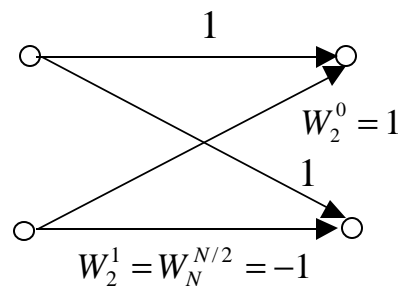
Notice that there is actually no need to perform any multiplication. For simplicity though, we assume the complexity of this butterfly is the same as the one that we presented earlier, i.e. 1 CM and 2 CAs.

$x[0]$ W_N^0 W_N^0 $X[0]$ W_N^0

$x[4]$ W_N^4 W_N^2 $X[1]$ W_N^1

$x[2]$ W_N^0 W_N^4 $X[2]$ W_N^2

$x[6]$ W_N^4 W_N^6 $X[3]$ W_N^3

$x[1]$ W_N^0 W_N^0 $X[4]$ W_N^4

$x[5]$ W_N^4 W_N^2 $X[5]$ W_N^5

$x[3]$ W_N^0 W_N^4 $X[6]$ W_N^6

$x[7]$ W_N^4 W_N^6 $X[7]$ W_N^7

$(m-1)$st stage W_N^r mth stage $W_N^{(r+N/2)}$

- It is evident that when $N = 2^v$, there will be altogether v stages in the overall computational structure, with $N/2$ butterflies at each stage.

The total number of complex multiplications is thus

$$n_M = v \times \frac{N}{2} \times 1 = \frac{1}{2} N \log_2(N)$$

The total number of additions is

$$n_A = 2 n_M = N \log_2(N).$$

In conclusion, the complexity of this decimation-in-time algorithm is in the order of $N \log N$, which is much smaller than N^2 for large N.

- It is observed from the $N=8$ example that the input data needed to be shuffled before being any computation can be take place. This can be done through the bit-reversal algorithm.

  Let

  $$\mathbf{A} = \big(a[0], a[1], ..., a[n], ..., a[N-1]\big)$$

  be the array containing the shuffled data. In the $N=8$ example, this vector is $\mathbf{A} = \big(x[0], x[4], x[2], x[6], x[1], x[5], x[3], x[7]\big)$. Furthermore, let

  $$\mathbf{B} = \big(b_{v-1}, b_{v-2}, ..., b_0\big); \qquad b_j \in \{0,1\}$$

  be the natural binary representation of the integer n, i.e.

  $$n = \sum_{j=0}^{v-1} b_j 2^j$$

  Then

  $$a[n] = x[m]$$

where

$$m = \sum_{j=0}^{v-1} b_j 2^{v-1-j}$$

is the decimal equivalent of the binary vector

$$\mathbf{B}_r = \left(b_0, b_1, ..., b_{v-1} \right).$$

- **Exercise**: Show that if $a[n] = x[m]$, then $a[m] = x[n]$, i.e. the samples are pair-wise interchanged.

- *In-place computations*

  The FFT algorithm computes the DFT coefficients in stages. The input data at the $(m\text{-}1)$-th stage ($m = 0$ to v from left to right) are transformed into another set of data that will be used in the computations in the m-th stage.

  Clearly, we can store the data to be computed in one array and those used in the computation in another array. This requires a total of $2N$ storage.

  However, the FFT algorithm we just described can be used to implement in-place computations. This will essentially reduce the storage to about 50%.
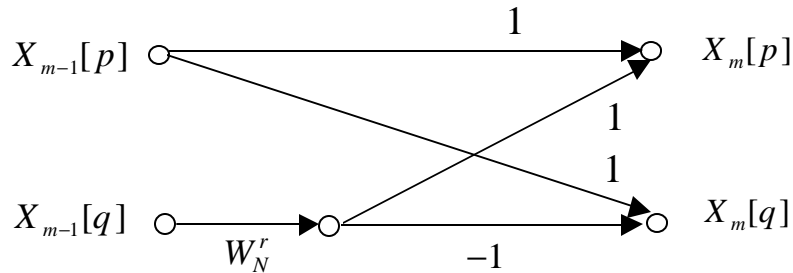
  Let us first index the data at the m-th stage in the signal flow graph as $X_m[\ell]$, where $\ell = 0, 1, ..., N-1$ (from top to bottom), then it can be easily shown that the pair

$$\left(X_m[p], X_m[q] \right)$$

can be computed exclusively from the pair

$$\left(X_{m-1}[p], X_{m-1}[q] \right)$$

using the butterfly

This means as far as data storage is concerned, we need only one array of N samples (and a temporary buffer of 2 samples). At the beginning of the m-th stage, this array stores all the $X_{m-1}[\ell]$, $\ell = 0,1,...,N-1$. As soon as $X_m[p]$ and $X_m[q]$ are computed, the corresponding entries in the array are then updated.

- There are other versions of FFT that do not require re-ordering of the input; see Fig. 9.14-9.16 of Text. Some of these algorithms, however, have the disadvantage that computations can not be implemented in place. Others, on the other hand, requires re-ordering of the output (see for example the decimation-in-frequency FFT algorithm described in Section 9-4).

- Algorithms for general value of N may have more complicated indexing problem.

4.3 Application of FFT in OFDM

- *References*:

  1. Chapter 12 of the book by J. G. Proakis, *Digital Communicaitons*, McGraw Hill, 4th Edition, 2001.

  2. IEEE Standard 802.11a-1999, Part 11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – High Speed Physical Layer in the 5 GHz Band.*

  3. ADSL Tutorial at

     http://www.rad.com/networks/1997/adsl/AdslMainPage.htm

- Orthogonal Frequency Division Multiplexing (OFDM) is a parallel modulation format. Instead of transmitting data serially at a high speed with a single carrier, an OFDM modem organizes the data into N lower-rate substreams and transmit the data in the different substreams at different carrier frequencies.
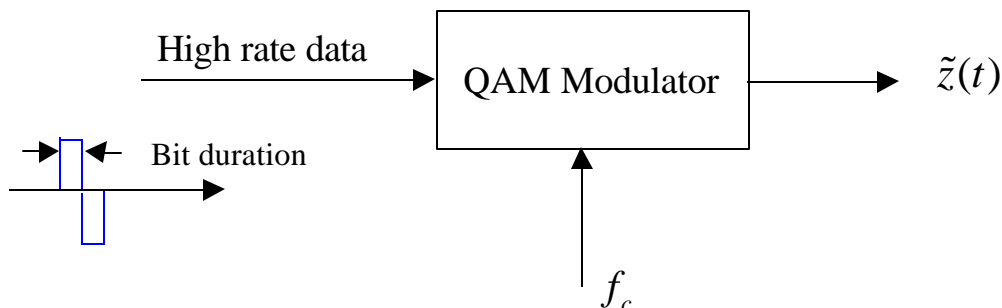
Conventional serial modem
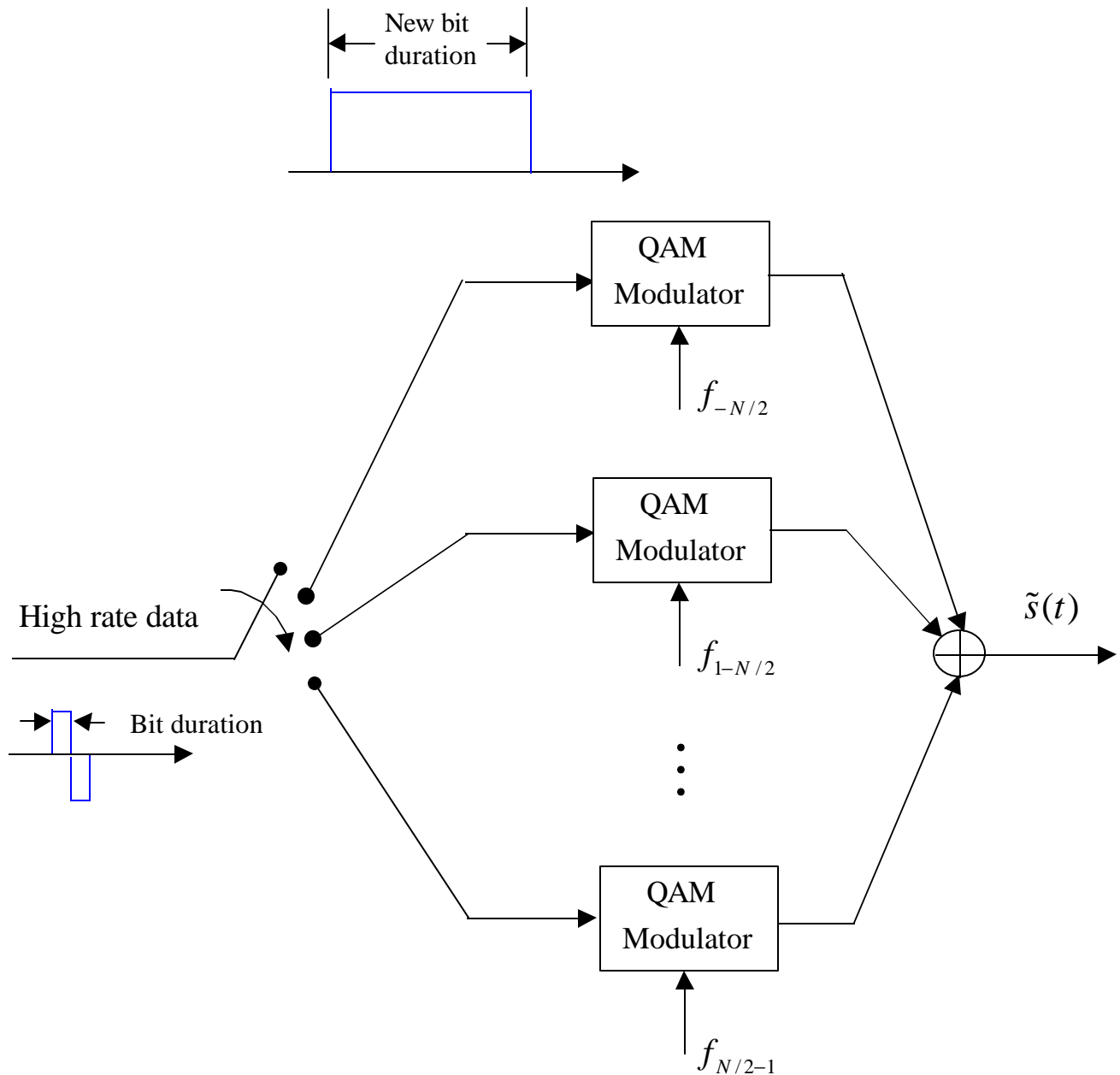
OFDM Modulator

- note that by a QAM modulator, we mean a modulator that generates a modulated signal of the form

$$w(t) = u(t) \cos\left(2\pi f_o t\right) - v(t) \sin\left(2\pi f_o t\right)$$

from the input data. Here $u(t)$ and $v(t)$ are baseband pulse-amplitude modulated (PAM) signals. Example of a QAM modulator is QPSK.

- Spectral characteristics of conventional serial modulation and OFDM
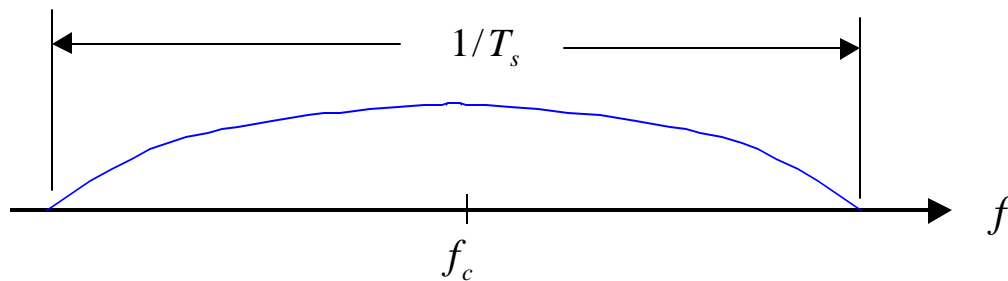
Spectral characteristics of conventional serial modem. T_s **is the symbol duration of the signal** $\tilde{z}(t)$**.**
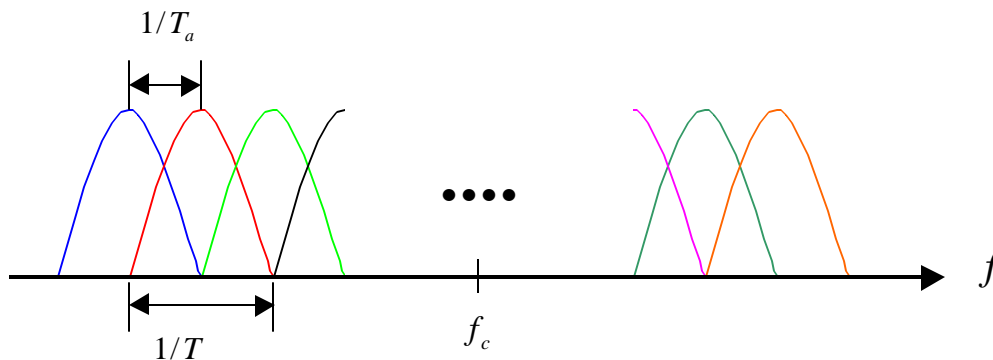
Spectral characteristics of the OFDM signal $\tilde{s}(t)$**.** T **is the symbol duration in the subchannels and** $1/T_a$ **is the frequency separation between adjacent subchannels.**

- Why OFDM?

  In many applications, the communication channel is not ideal, i.e. its frequency response is not flat (nor linear phase) over the transmission band.

  A nonideal channel causes severe intersymbol interference (ISI) in serial modems.

  By dividing the entire transmission band into subchannels, then as long as the frequency response over a subchannel is flat (and linear phase), that subchannel is ideal.

- The subchannel spacing (and the number of subchannels) is thus a function of the coherent bandwidth of the channel.

4.3.1 The Basic OFDM Signal Structure

- Data in an OFDM system is organized into frames. In the basic OFDM system, each frame is of duration T_a sec, where $1/T_a$ is the symbol rate of each subchannel.

- Without loss in generality, we can focus on the frame in the interval $[0, T_a]$. The OFDM signal in this interval can be written as

$$\tilde{s}(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \tilde{s}_k(t); \qquad 0 \le t \le T_a$$

  where

$$\tilde{s}_k(t) = X[k]\cos\left(2\pi f_k t\right) - Y[k]\sin\left(2\pi f_k t\right)$$

is the output signal of the k-th QAM modulator. In the above equation,

$$X[k] \in \{\pm 1, \pm 3, \ldots, \pm(Q-1)\}$$

and

$$Y[k] \in \{\pm 1, \pm 3, \ldots, \pm(Q-1)\}$$

are respectively the in-phase (I) and quadrature (Q) channel data of the k-th modulator in the interval $[0, T_a]$,

$$M = Q^2$$

is the number of points in the signal constellation, and

$$f_k = f_c + \frac{k}{T_a}$$

is the k-th subcarrier frequency. The term f_c is the center frequency of the OFDM signal.

- The *signal constellation* of the k-th modulator is obtained by plotting all possible

$$S[k] = X[k] + jY[k]$$

as points in the complex plane; see the examples below.

It is possible to use different constellations in different subchannels. A "good" subchannel can use a denser (and hence more efficient) constellation while a "bad" channel uses a less dense constellation.

BPSK   Q

b_0

+1

0   1
-1   +1   I

-1

QPSK   Q b_0b_1

01   +1   11

-1   +1   I
00   10
-1

16-QAM   Q $b_0b_1b_2b_3$

00 10   01 10   11 10   10 10
+3

00 11   01 11   11 11   10 11
+1

-3   -1   +1   +3   I

00 01   01 01   11 01   10 01
-1

00 00   01 00   11 00   10 00
-3

64-QAM   Q $b_0b_1b_2b_3\,b_4b_5$

000 100   001 100   011 100   010 100 | 110 100   111 100   101 100   100 100
+7

000 101   001 101   011 101   010 101 | 110 101   111 101   101 101   100 101
+5

000 111   001 111   011 111   010 111 | 110 111   111 111   101 111   100 111
+3

000 110   001 110   011 110   010 110 | 110 110   111 110   101 110   100 110
+1

-7   -5   -3   -1   +1   +3   +5   +7   I

000 010   001 010   011 010   010 010 | 110 010   111 010   101 010   100 010
-1

000 011   001 011   011 011   010 011 | 110 011   111 011   101 011   100 011
-3

000 001   001 001   011 001   010 001 | 110 001   111 001   101 001   100 001
-5

000 000   001 000   011 000   010 000 | 110 000   111 000   101 000   100 000
-7

Signal constellations of BPSK, QPSK, 16QAM and 64QAM. The binary patterns in these constellations are the information contained in the signal points.

- **Exercise**: Show that the overall bit rate of the above OFDM signal is

$$R_b = N \times \log_2 M \times \frac{1}{T_a}$$

- **Exercise**: Convince yourself that when $Q=2$, the signal in each subchannel is a QPSK signal with rectangular pulse shaping.

- Consider the pairwise correlation of the subchannel signals:

$$\int_0^{T_a} \tilde{s}_k(t)\tilde{s}_m(t)\,dt = \int_0^{T_a} \left\{ X[k]\cos(2\boldsymbol{p}\,f_k t) - Y[k]\sin(2\boldsymbol{p}\,f_k t) \right\}$$

$$\left\{ X[m]\cos(2\boldsymbol{p}\,f_m t) - Y[m]\sin(2\boldsymbol{p}\,f_m t) \right\} dt$$

$$= \int_0^{T_a} X[k]X[m]\cos(2\boldsymbol{p}\,f_k t)\cos(2\boldsymbol{p}\,f_m t)\,dt$$

$$+ \int_0^{T_a} Y[k]Y[m]\sin(2\boldsymbol{p}\,f_k t)\sin(2\boldsymbol{p}\,f_m t)\,dt$$

$$- \int_0^{T_a} X[k]Y[m]\cos(2\boldsymbol{p}\,f_k t)\sin(2\boldsymbol{p}\,f_m t)\,dt$$

$$- \int_0^{T_a} Y[k]X[m]\sin(2\boldsymbol{p}\,f_k t)\cos(2\boldsymbol{p}\,f_m t)\,dt$$

Since

$$\int_0^{T_a} \cos\left(2\mathbf{p}\, f_k t\right)\cos\left(2\mathbf{p}\, f_m t\right)dt = \frac{1}{2}\int_0^{T_a}\left\{\cos\left(2\mathbf{p}[\,f_k + f_m]t\right)+\cos\left(2\mathbf{p}[\,f_k - f_m]t\right)\right\}dt$$

$$= \frac{1}{2}\int_0^{T_a}\left\{\cos\left(4\mathbf{p}\, f_c t + 2\mathbf{p}\left[\tfrac{k+m}{T_a}\right]t\right)+\cos\left(2\mathbf{p}\left[\tfrac{k-m}{T_a}\right]t\right)\right\}dt$$

$$= \begin{cases} T_a/2 & k = m \\ 0 & k \neq m \end{cases},$$

$$\int_0^{T_a} \sin\left(2\mathbf{p}\, f_k t\right)\sin\left(2\mathbf{p}\, f_m t\right)dt = \frac{1}{2}\int_0^{T_a}\left\{\cos\left(2\mathbf{p}[\,f_k - f_m]t\right)-\cos\left(2\mathbf{p}[\,f_k + f_m]t\right)\right\}dt$$

$$= \frac{1}{2}\int_0^{T_a}\left\{\cos\left(2\mathbf{p}\left[\tfrac{k-m}{T_a}\right]t\right)-\cos\left(4\mathbf{p}\, f_c t + 2\mathbf{p}\left[\tfrac{k+m}{T_a}\right]t\right)\right\}dt$$

$$= \begin{cases} T_a/2 & k = m \\ 0 & k \neq m \end{cases},$$

and

$$\int_0^{T_a} \sin\left(2\mathbf{p}\, f_k t\right)\cos\left(2\mathbf{p}\, f_m t\right)dt = \frac{1}{2}\int_0^{T_a}\left\{\sin\left(2\mathbf{p}[\,f_k - f_m]t\right)+\sin\left(2\mathbf{p}[\,f_k + f_m]t\right)\right\}dt$$

$$= \frac{1}{2}\int_0^{T_a}\left\{\sin\left(2\mathbf{p}\left[\tfrac{k-m}{T_a}\right]t\right)+\sin\left(4\mathbf{p}\, f_c t + 2\mathbf{p}\left[\tfrac{k+m}{T_a}\right]t\right)\right\}dt$$

$$= 0$$

when $f_c \gg 1/T_a$, the correlation becomes

$$\int_0^{T_a} \tilde{s}_k(t)\,\tilde{s}_m(t)\,dt = \begin{cases} \left(X^2[k]+Y^2[k]\right)T_a/2 & k = m \\ 0 & k \neq m \end{cases}.$$

In other word, the signals in different subchannels are orthogonal to one another and hence the term OFDM.

- To ensure orthogonality, the different QAM modulators must be in phase-synchronsim. This is because

$$\int_0^{T_a} \{X[k]\cos(2\pi f_k t + \phi_k) - Y[k]\sin(2\pi f_k t + \phi_k)\} \times$$

$$\{X[m]\cos(2\pi f_m t + \phi_m) - Y[m]\sin(2\pi f_m t + \phi_m)\} dt \neq 0$$

if the phases ϕ_k and ϕ_m are different. Unfortunately, it will be very complicated and expensive to keep the different modulators in phase-sync using analog technologies.

A DSP approach, based on FFT, can get around the problem.

- Consider the k-th subchannel signal again. This signal can be written as

$$\tilde{s}_k(t) = X[k]\cos(2\pi f_k t) - Y[k]\sin(2\pi f_k t)$$

$$= \text{Re}\{(X[k] + jY[k])(\cos(2\pi f_k t) + j\sin(2\pi f_k t))\}$$

$$= \text{Re}\{S[k]\exp(j2\pi f_k t)\}$$

$$= \text{Re}\left\{S[k]\exp\left(j2\pi\left[\frac{k}{T_a}\right]t\right)\exp(j2\pi f_c t)\right\}$$

$$= \text{Re}\{s_k(t)\exp(j2\pi f_c t)\}$$

where

$$s_k(t) = S[k]\exp\left(j2\pi\frac{k}{T_a}t\right)$$

is the baseband equivalent (or complex envelop) of $\tilde{s}_k(t)$.

- The OFDM signal can now be written as

$$\tilde{s}(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \tilde{s}_k(t)$$

$$= \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \mathrm{Re}\left\{s_k(t)\exp\left(j2\pi f_c t\right)\right\}$$

$$= \mathrm{Re}\left\{\left(\frac{1}{N} \sum_{k=-N/2}^{N/2-1} s_k(t)\right)\exp\left(j2\pi f_c t\right)\right\}$$

$$= \mathrm{Re}\left\{s(t)\exp\left(j2\pi f_c t\right)\right\}$$

where

$$s(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} s_k(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} S[k]\exp\left(j2\pi \frac{k}{T_a} t\right)$$

$$= u(t) + jv(t)$$

is the complex envelop of $\tilde{s}(t)$. This complex envelop has a real part of $u(t)$ and an imaginary part of $v(t)$.

- In a DSP implementation of the OFDM modem, the complex envelop $s(t)$ is generated digitally using a DSP and DAC. The two components of this signal is then fed to a QAM modulator operating at a center frequency of f_c; see diagrams next page.

  Since there is only one QAM modulator, there is no need to worry about relative phase shifts.
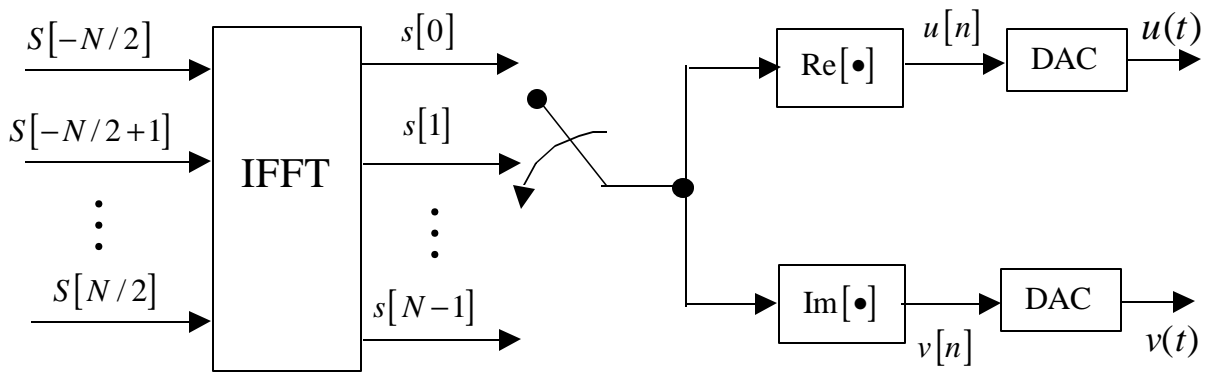
Baseband segment of the OFDM modulator
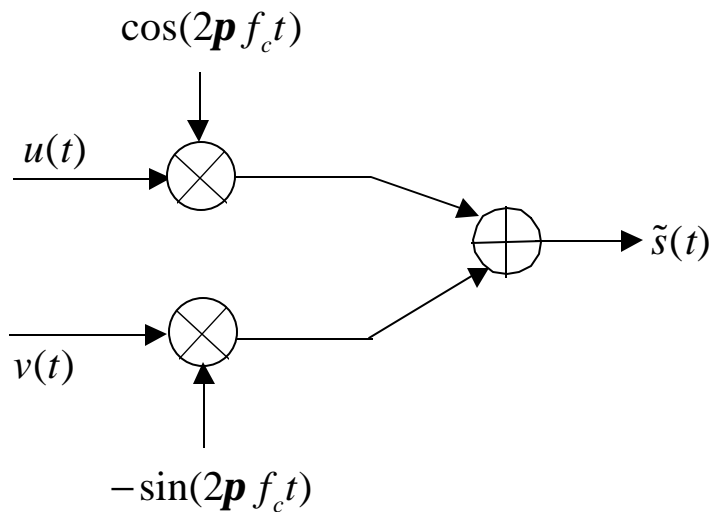
IF/RF segment of OFDM modulator

- Let

$$s[n] = s(t)\big|_{t=nT_a/N}$$

be the DT signal obtained by sampling the baseband OFDM signal $s(t)$ at time $t = nT_a/N$; $n = 0,1,...,N-1$. This means

$$s[n] = \left[\frac{1}{N}\sum_{k=-N/2}^{N/2-1} S[k]\exp\left(j2\boldsymbol{p}\frac{k}{T_a}t\right)\right]_{t=nT_a/N}$$

$$= \frac{1}{N}\sum_{k=-N/2}^{N/2-1} S[k]\exp\left(j2\boldsymbol{p}\frac{k}{T_a}\frac{nT_a}{N}\right)$$

$$= \frac{1}{N}\sum_{k=-N/2}^{N/2-1} S[k]W_N^{-kn}$$

$$= \frac{1}{N}\sum_{k=0}^{N/2-1} S[k]W_N^{-kn} + \frac{1}{N}\sum_{k=-N/2}^{-1} S[k]W_N^{-kn}$$

$$= \frac{1}{N}\sum_{k=0}^{N/2-1} S[k]W_N^{-kn} + \frac{1}{N}\sum_{k=N/2}^{N-1} S[k-N]W_N^{-(k-N)n}$$

$$= \frac{1}{N}\sum_{k=0}^{N/2-1} S[k]W_N^{-kn} + \frac{1}{N}\sum_{k=N/2}^{N-1} S[k-N]W_N^{-kn}$$

$$= \underbrace{\frac{1}{N}\sum_{k=0}^{N-1} D[k]W_N^{-kn}}_{\text{IDFT of } D[k]}; \qquad\qquad n = 0,1,2,...,N-1$$
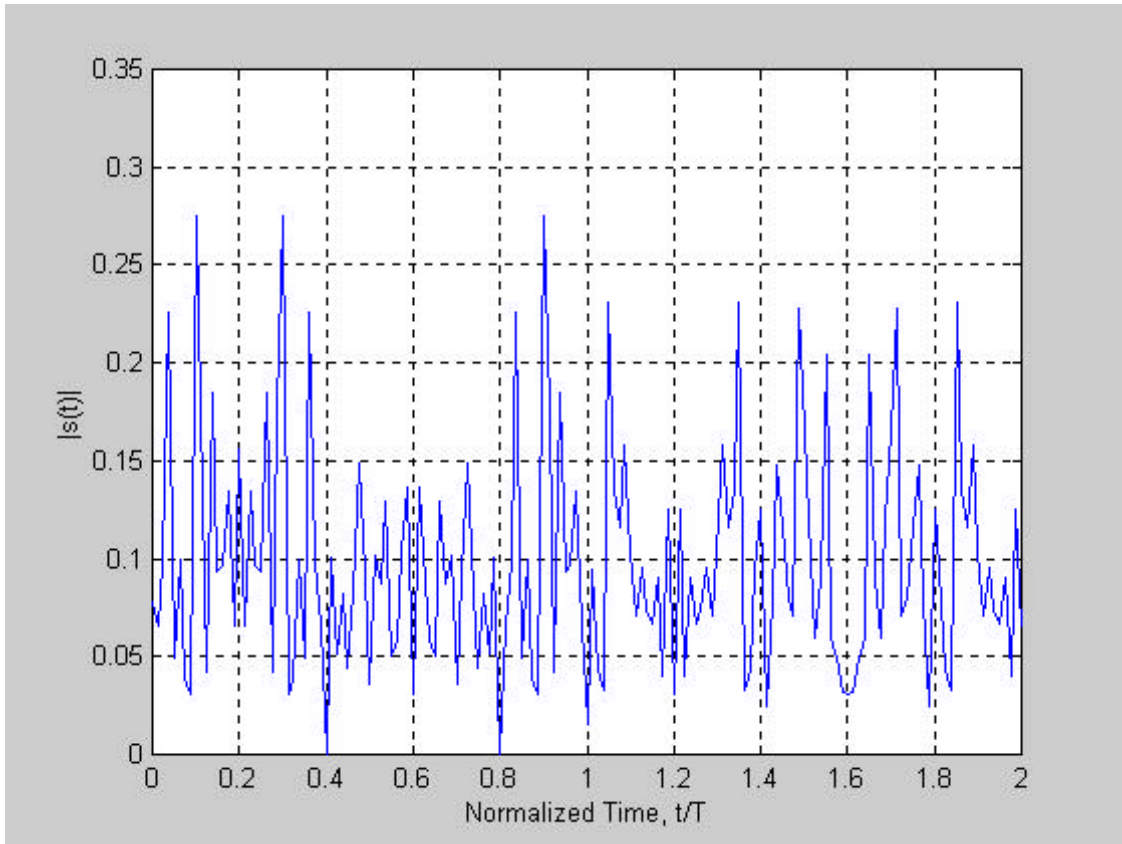
where

$$D[k] = \begin{cases} S[k] & k = 0,1,...,N/2-1 \\ S[k-N] & k = N/2,...,N-1 \end{cases}$$

represents the permutated data in the different subchannels.

At this point, it is obvious that we can exploit the efficiency of the FFT to compute the $s[n]$'s.

The signal $s(t)$ is obtained by passing the $s[n]$'s to a digital to analog converter (DAC).

- Example of an OFDM waveform over two symbol intervals:

Baseband OFDM waveform in wireless LAN: $N = 64$ but with only 53 active subchannels (the inactive channels have data values of 0); BPSK modulation in each subchannel; 20% guard interval ($T_a = 0.8T$). A normalized time of 1 corresponds to 1 symbol interval.

- It is observed that the magnitude of the signal fluctuates over a wide range, i.e. a large peak-to-average power ratio. This is a drawback of OFDM because many power amplifiers in wireless applications are peak power limited while the performance is determined by the average power.

- Clipping or waveform coding can be used to reduce the fluctuation.

- As for demodulation, let

$$r(t) = s(t) + n(t)$$

be the received signal. Here $n(t)$ represents channel noise. If the channel is noiseless, then $r(t) = s(t)$. It is obvious in this ideal case that if the receiver samples $r(t)$ at $t = nT_a / N$, $n = 0, 1, ..., N-1$, and then takes a FFT of the resultant DT signal, it will obtain the permutated data $D[k]$. In the presence of noise, the result simply becomes $D[k]$ plus a noise term.

4.3.2 Modified OFDM for Multipath

- While the OFDM signal described in Section 4.3.1 is sufficient for the additive noise channel, it offers little protection against multipath (which is an example of a non-ideal channel).

  If the channel is an additive noise channel, there is no point in using OFDM in the first place because of its large peak to average power ratio. A conventional (serial) modulation scheme will do better and will be simpler to implement.

- To be anti-multipath, the original OFDM signal should be modified to (again, focusing on one-frame only)

$$a(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} S[k] \exp\left(j2\boldsymbol{p} \frac{k}{T_a}(t - T_G) \right), \qquad 0 \leq t \leq T$$

where

$$T = T_a + T_G$$

is the new OFDM symbol period, with T_a being the *fundamental symbol duration*, and T_G being the duration of the *guard interval*.

The frequency-spacing between adjacent sub-channels is still $1/T_a$.

- The value of T_G should be at least the delay spread of the multipath channel. (In general, it should be the inverse of the coherent bandwidth of a non-ideal channel.)

- **Exercise**: Determine the bit rate of the above modified OFDM signal, assuming all the other system parameters remain the same .

- The modified OFDM signal $a(t)$ is divided into two segments, one in the interval $[0,T_G]$, and the other in the interval $[T_G,T]$. The latter segment is called the **effective data** while the former is called the **cyclic prefix** or **guard interval**. It should be clear that the effective data segment is

$$a(t) = s(t - T_G); \qquad T_G \leq t \leq T,$$

where $s(t)$ is the basic OFDM signal in Section 4.3.1. The cyclic prefix is simply a repeat of the last portion of the effective data; see figure below.
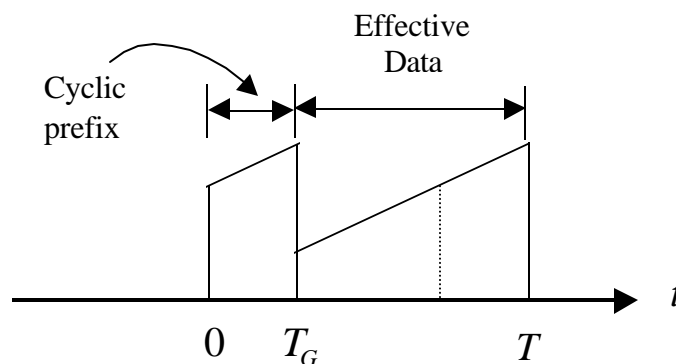
- Samples of the effective data segment of $a(t)$ are obtained via an IFFT of the permutated data $D[k]$; see Section 4.3.1. The resultant DT signal is again denoted by $s[n]$; $n = 0,1,...,N-1$. However $s[n]$ is now the value of the modified OFDM signal $a(t)$ at $t = T_G + (n-1)T_a / N$.

- Assuming

$$T_G = L\left(T_a / N\right)$$

(i.e. the Guard interval is made up of L samples) and let

$$a[n] = a(t)\Big|_{t=nT_a/N}; \qquad n = 0,1,..., N+L-1$$

This means for the effective data section,

$$a[n] = s\big[n-L\big]; \qquad n = L, L+1,..., L+N-1.$$

As for the cyclic prefix, we have

$$a[n] = s\big[N-L+n\big]; \qquad n = 0,1,..., L-1.$$

- Let $r(t)$ be the received signal when the modified OFDM signal is transmitted over a communication channel consisting of exactly L paths with path gains

$$g_0, g_1,..., g_{L-1}$$

and delays

$$0, \; \Delta t, \; 2\Delta t, \ldots, \; (L-1)\Delta t \; .$$

Here

$$\Delta t = T_a / N$$

is the time-separation between adjacent samples in the DT signal $a[n]$. It can be shown that the following row sum yields the samples of $r(t)$ at $t = n\Delta t$ (only one symbol is shown)

$$
\begin{aligned}
&g_0 s[N-L], g_0 s[N-L+1], \ldots, \; g_0 s[N-1], \| \, g_0 s[0], && g_0 s[1], \; \ldots, && g_0 s[N-1] \| \\
&\qquad g_1 s[N-L], \quad \ldots, \; g_1 s[N-2], \| \, g_1 s[N-1], && g_1 s[0], \; \ldots, \; g_1 s[N-2], \| \, g_1 s[N-1] \\
&\qquad\qquad \ddots \qquad\qquad \ddots \quad \ddots \qquad \ddots \qquad\qquad \ddots \qquad \ddots \\
&\qquad\qquad g_{L-1} s[N-L], \| \, g_{L-1} s[N-L+1], \ldots, && g_{L-1} s[N-L], \| \, g_{L-1} s[N-L+1], \ldots, g_{L-1} s[N-1]
\end{aligned}
$$

where the sign "$\| \bullet \|$" are the delimiters for the time interval $[T_G, T]$. Let the received samples in this interval be denoted by $r[n]$, $n = 0, 1, \ldots, N-1$, and define the vector $\mathbf{r}$ as

$$
\mathbf{r} = \begin{bmatrix} r[0] \\ r[1] \\ \vdots \\ r[N-1] \end{bmatrix}.
$$

It is observed that

$$r[n] = \sum_{m=0}^{L-1} g_m s\left[(n-m)_N \right]; \qquad n = 0, 1, \ldots, N-1$$

where $(\bullet)_N$ represents a mod-N operation. This summation is a circular convolution and is vital to the anti-multipath capability of OFDM.

- In matrix form, the circular convolution can be written as

$$\mathbf{r} = \sum_{m=0}^{L-1} g_m \mathbf{P}^m \mathbf{s} = \left(\sum_{m=0}^{L-1} g_m \mathbf{P}^m \right) \mathbf{W}^{-1} \mathbf{D}$$

where

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

is a cyclic permutation matrix,

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{(N-1)2} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

is the matrix representing DFT with

$$\mathbf{W}^{-1} = \frac{1}{N} \mathbf{W}^{\dagger},$$

and

$$\mathbf{D} = \begin{bmatrix} D[0] \\ D[1] \\ \vdots \\ D[N-1] \end{bmatrix},$$

is the permutated data vector.

- To detect the data vector $\mathbf{D}$, the receiver performs a DFT of the $r[n]$ s. This is equivalent to computing

$$\mathbf{R} = \mathbf{Wr}$$

$$= \mathbf{W}\left(\sum_{m=0}^{L-1} g_m \mathbf{P}^m\right)\mathbf{W}^{-1}\mathbf{D}$$

$$= \left(\sum_{m=0}^{L-1} g_m \mathbf{W}\mathbf{P}^m\mathbf{W}^{-1}\right)\mathbf{D}$$

$$= \left(\sum_{m=0}^{L-1} g_m \mathbf{Q}_m\right)\mathbf{D}$$

where (show this)

$$\mathbf{Q}_m = \mathbf{W}\mathbf{P}^m\mathbf{W}^{-1} = \frac{1}{N}\mathbf{W}\mathbf{P}^m\mathbf{W}^\dagger = \begin{bmatrix} W_N^0 & & & & \\ & W_N^m & & & \\ & & W_N^{2m} & & \\ & & & \ddots & \\ & & & & W_N^{(N-1)m} \end{bmatrix}$$

is a diagonal matrix. Consequently, the recovered data can be rewritten as

$$\mathbf{R} = \left(\sum_{m=0}^{L-1} g_m \mathbf{Q}_m\right)\mathbf{D} = \mathbf{HD},$$

where

$$\mathbf{H} = \sum_{m=0}^{L-1} g_m \mathbf{Q}_m = \begin{bmatrix} H_0 & & & \\ & H_1 & & \\ & & \ddots & \\ & & & H_{N-1} \end{bmatrix}$$

is also a diagonal matrix whose k-th element,

$$H_k = \sum_{m=0}^{L-1} g_m W_N^{km} \; ,$$

represents the frequency response of the channel at the subcarrier frequency

$$f_k = f_c + \frac{k}{T_a}$$

- Since the matrix $\mathbf{H}$ is diagonal, the k-th recovered data is simply

$$R[k] = H_k D[k].$$

The result implies that there is no ISI, even though the channel is a non-ideal channel.

- It should be pointed out that in order to successfully detect the data in $R[k]$, the frequency response H_k needs to be estimated. This can be done by inserting a preamble (training symbols) to the data. Alternatively, we can use differential detection.