EE 512

Digital Signal Processing

Session 11

October 6, 1992

M. Azimi 800-525-4950, ext. 7956

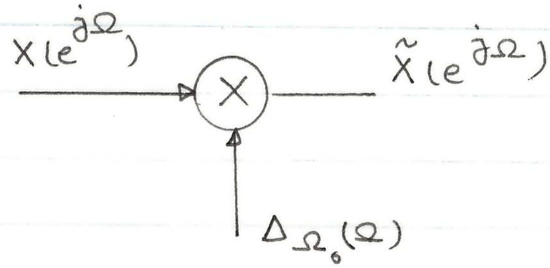
Colorado State University

Frequency Domain Sampling of time-limited Signals and Discrete Fourier Series

Let $x(n)$ be time-limited signal $x(n) = 0 \quad n < 0 \;, n > N-1$ and assume that the DTFT of $x(n)$ is $X(\Omega)$ or $X(e^{j\Omega})$. Now let us sample $X(e^{j\Omega})$ to obtain the sampled spectrum $\tilde{X}(e^{j\Omega})$

$$\tilde{X}(e^{j\Omega}) = X(e^{j\Omega})\, \Delta_{\Omega_0}(\Omega)$$

or $$\tilde{X}(\Omega) = X(\Omega)\, \Delta_{\Omega_0}(\Omega)$$



where
$$\Delta_{\Omega_0}(\Omega) = \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_0)$$

Ω_0 = Spacing Between frequency samples

with inverse DTFT that is

$$\Delta_N(n) = \frac{1}{\Omega_0} \sum_{r=-\infty}^{\infty} \delta(n - rN) \;, \qquad N = \frac{2\pi}{\Omega_0}$$

Then
$$\tilde{X}(\Omega) = X(\Omega)\, \Delta_{\Omega_0}(\Omega)$$
$$= X(\Omega) \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_0)$$
$$= \sum_{k=-\infty}^{\infty} X(k\Omega_0)\, \delta(\Omega - k\Omega_0)$$
$$= \{ X(k\Omega_0) \} \;, \forall k \qquad X(k\Omega_0) = \sum_{n=0}^{N-1} x(n)\, e^{-j\Omega_0 k n}$$

If $\frac{2\pi}{\Omega_0} = N$ (an integer) then

$$X(k\Omega_0) \triangleq X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-2\pi j \frac{nk}{N}} \quad , \forall k \in [0, N-1]$$

$$= \underset{\text{coeff}}{\text{DFS}} \{ x(n) \}$$

i.e. $X(k)$'s for $k \in [0, N-1]$ are the Discrete Fourier Coefficients functions.

$$D_n(k+rN) = e^{-j2\pi n(k+rN)/N} = e^{-j2\pi \frac{nk}{N}} \underbrace{e^{-j2\pi nk}}_{1}$$

$$= e^{-j2\pi \frac{nk}{N}} = D_n(k)$$

i.e. $D_n(0) = D_n(N)$, $D_n(1) = D_n(N+1)$, ... etc.

Also $D_{n+\ell N}(k) = e^{-j2\pi(n+\ell N)k/N} = e^{-j2\pi nk/N} = D_n(k)$

2- The base functions $D_n(k)$ for $k, n \in [0, N-1]$ form a complete orthogonal space

$$\frac{1}{N}\sum_{k=0}^{N-1} D_m(k) D_n^*(k) = \frac{1}{N}\sum_{k=0}^{N-1} e^{j2\pi \frac{k}{N}(n-m)}$$

$$= \begin{cases} 1 & m = n \\ \dfrac{1 - e^{j2\pi k(n-m)N/N}}{1 - e^{j2\pi k(n-m)/N}} = 0 & m \neq n \end{cases}$$

i.e.

$$\frac{1}{N}\sum_{k=0}^{N-1} D_m(k) D_n^*(k) = \delta(m-n)$$

Using this orthogonality property

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

$$= \sum_{m=0}^{N-1} x(m)\,\delta(m-n) = x(n)$$

Thus $x(n)$ can be recovered from its DFS coefficients using

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j2\pi \frac{kn}{N}}$$

i.e.

$$x(n) \xleftrightarrow{DFS} X(k)$$ one-to-one correspondence

3- DFS is used to represent periodic discrete-time signals.

4- Both $X(k)$ and $x(n)$ are periodic with period N.

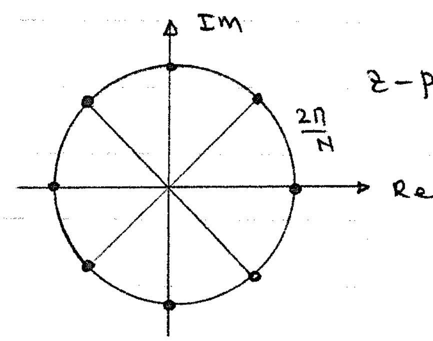
5- In contrast to the continuous-time case, there are no convergence issues and no Gibbs phenomenon. The reason being that any discrete-time periodic sequence $x(n)$ is completely specified by N (finite) number of base functions hence there will be no truncation.

6- Recall

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \quad \text{or} \quad X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} \quad \text{for finite duration signals}$$

and

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}$$



Thus

$$X(k) = X(z)\Big|_{z=e^{j2\pi k/N}}$$

This corresponds to sampling the z-Transform $X(z)$ at N-point equally spaced in angle around the unit circle with 1st sample occuring at $z=1$.

7) As in the continuous-time case, when the period of the sequence in the discrete-time domain increases, the samples become more and more finely spaced.

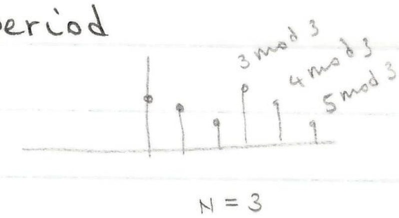
Fourier Representation of Finite-Duration Sequences and Discrete Fourier Transform

Consider a finite duration sequence $x(n)$ of length N

$$x(n) = 0 \qquad n < 0 \; , \; n > N-1$$

Let $\tilde{x}(n)$ be the periodic replication of $x(n)$ i.e.

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n + rN) \qquad N : \text{period}$$



$$= x(n \text{ modulo } N)$$

$$= x((n))_N$$

Note: with n expressed as $n = n_1 + \overset{r}{n_2} N$, $\quad n_1 \in [0, N-1]$

$n \bmod N = n_1$. The finite duration sequence $x(n)$ is obtained from $\tilde{x}(n)$ by extracting one period i.e.

$$x(n) = \begin{cases} \tilde{x}(n) & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

$x(n) = \tilde{x}(n) R_N(n)$

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

The DFS Coefficients of $\tilde{x}(n)$ are

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) \, W_N^{kn} \quad , \quad W_N \triangleq e^{-j\frac{2\pi}{N}}$$

As was shown $\tilde{X}(k)$ is periodic with period N. The Fourier Coeff. associated with a finite duration signal will also be a finite duration sequence Corresponding to one-period of $\tilde{X}(k)$ i.e.

$$X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n) \, W_N^{kn} & k \in [0, N-1] \\ 0 & \text{otherwise} \end{cases}$$

$$= \text{DFT}\{x(n)\}_{k \in [0, N-1]} \qquad \text{Discrete Fourier transform}$$

Also

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-kn}$$

and

$$x(n) = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} & n \in [0, N-1] \\ 0 & \text{otherwise} \end{cases}$$

$$= \text{IDFT}\{X(k)\} \quad n \in [0, N-1] \quad \text{Inverse DFT}.$$

There exist extremely fast algorithms such as FFT, WFT for the computation of DFT and IDFT of finite duration sequences.

Properties of DFT

1- Linearity

Let

$$x_3(n) = a\, x_1(n) + b\, x_2(n) \qquad \begin{matrix} x_1(n) = 0 & n < 0,\ n > N_1 - 1 \\ x_2(n) = 0 & n < 0,\ n > N_2 - 1 \end{matrix}$$

then

$$X_3(k) = a\, X_1(k) + b\, X_2(k)$$

If $x_1(n)$ is of size N_1 and $x_2(n)$ is of size N_2, then the max duration of $x_3(n)$ is $N_3 = \text{Max}[N_1, N_2]$ and the DFTs must be computed with $N = N_3$. e.g. if $N_1 < N_2$ then $x_1(n)$ is augmented by $N_2 - N_1$ zeros and we get

$$X_1(k) = \sum_{n=0}^{N_1-1} x_1(n) W_{N_2}^{kn} \qquad k \in [0, N_2-1]$$

$$= \text{DFT}\{x_1(n)\} \text{ of size } N_2$$

$$X_2(k) = \sum_{n=0}^{N_2-1} x_2(n) W_{N_2}^{kn} \qquad k \in [0, N_2-1]$$

$$= \text{DFT}\{x_2(n)\} \text{ of size } N_2$$

EE 512

Digital Signal Processing

Session 12

October 8, 1992

M. Azimi 800-525-4950, ext. 7956

Colorado State University

EE 512

Digital Signal Processing

Session 14

October 15, 1992

Ass 3
4.10, 4.15, 4.18, 4.28,
4.35, 4.81, 5.4(a)
5.6, 5.11, 5.18,
Due october 17th

M. Azimi 800-5

Co

Uni

Example on DFT (6.19)

Find DFT of (a) $\delta(n-n_0)$ (b) a^n $n \in [0, N-1]$

(c) $e^{j(\frac{2\pi}{N})k_0 n}$ $n \in [0, N-1]$

(d) $\cos \frac{2\pi}{N} k_0 n$ $n \in [0, N-1]$

(e) $x(n) = \begin{cases} 1 & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$ $n \in [0, N-1]$

(a) $$X(k) = \sum_{n=0}^{N-1} \delta(n-n_0)\, e^{-j 2\pi/N\, kn} = e^{-j\frac{2\pi}{N} k n_0}$$ $k \in [0, N-1]$

(b) $$X(k) = \sum_{n=0}^{N-1} a^n e^{-j\frac{2\pi}{N} nk} = \sum_{n=0}^{N-1} (a e^{-j\frac{2\pi}{N}k})^n$$

$$= \frac{1-a^N}{1-a e^{-j2\pi k/N}}$$

(c) $$X(k) = \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N} nk_0} e^{-j\frac{2\pi}{N} nk} = N\,\delta(k-k_0)$$

(d) $$X(k) = \sum_{n=0}^{N-1} \frac{1}{2}\left[e^{j\frac{2\pi}{N} nk_0} + e^{-j\frac{2\pi}{N} nk_0}\right] e^{-j\frac{2\pi}{N} nk}$$

$$= \frac{N}{2}\left[\delta(k-k_0) + \delta(k-N+k_0)\right]$$

(e) $$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N} nk}$$ N : Assume odd

$$= 1 + e^{-j\frac{2\pi}{N}2k} + e^{-j\frac{2\pi}{N}4k} + \dots e^{-j\frac{2\pi}{N}(N-1)k}$$

$$= \frac{1 - e^{-j\frac{2\pi}{N}2k\left(\frac{N+1}{2}\right)}}{1-e^{-j2\pi/N\, 2k}} = \frac{1-e^{-j\frac{2\pi}{N}k}}{1-e^{-j4\pi k/N}}$$

$$= \frac{1}{1+e^{-j\frac{2\pi}{N}k}}$$

2- Circular Shift of a Sequence

$x(n) = 0 \quad n < 0 \;,\; n > N-1$

$\tilde{x}(n) = x((n))_N$

Suppose $\tilde{x}(n)$ is the periodic extension of $x(n)$ and $\tilde{x}(n+m)$ is shifted $\tilde{x}(n)$ by m samples. Then the shifted version of $x(n)$ is obtained by extracting one period of $\tilde{x}(n+m)$ in the range $0 \leq n \leq N-1$

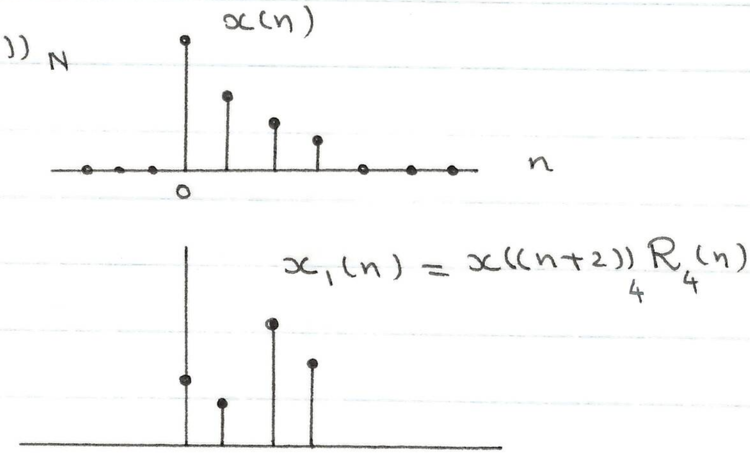
Let $\quad \tilde{x}_1(n) = \tilde{x}(n+m) = x((n+m))_N$

and $\quad x_1(n) = x((n+m))_N \, R_N(n)$

$\tilde{x}_1(n)$

where

$$R_N(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{elsewhere} \end{cases}$$



Also we have

$$DFS\{\tilde{x}(n+m)\} = W_N^{-km} \, DFS\{\tilde{x}(n)\} = W_N^{-km} \tilde{X}(k) = \tilde{X}_1(k)$$

then

$$X_1(k) = W_N^{-km} X(k) \quad , \quad X(k) = \tilde{X}(k) R_N(k)$$

Note that $\tilde{x}_1(n)$ is obtained by a linear shift of the periodic sequence $\tilde{x}(n)$, and $x_1(n)$ is obtained by a rotation of $x(n)$ signal samples around a cylinder. Thus such a shift is called "circular shift".

Because of duality between the time and frequency domains a similar result holds when a circular shift is applied to the DFT coefficients i.e. if

$$X_1(k) = X((k+l))_N \, R_N(k)$$

then

$$x_1(n) = W_N^{ln} \, x(n)$$

3- Circular Convolution

We saw that multiplication of the DFS coefficients of two sequences corresponds to a periodic convolution of the sequences. Now let us consider two finite duration sequences $x_1(n)$ and $x_2(n)$ with DFTs $X_1(k)$ and $X_2(k)$ we wish to determine

$$x_3(n) = \text{IDFT}\{X_1(k)\,X_2(k)\}$$

specifically, $x_3(n)$ corresponds to one period of $\tilde{x}_3(n)$ and is given by

$$x_3(n) = \left[\sum_{m=0}^{N-1} \tilde{x}_1(m)\,\tilde{x}_2(n-m)\right] R_N(n)$$

$$= \left[\sum_{m=0}^{N-1} x_1((m))_N\, x_2((n-m))_N\right] R_N(n)$$

$$= x_1(n) \bigcirc x_2(n)$$

i.e. multiplying the DFTs of two finite duration sequences and then taking IDFT of the result is equivalent to circularly convolving the sequences or linear convolution of the relevant periodic sequences and then extracting the region between $0 \leq n \leq N-1$.

Example

Let $x(n) = \{1, 2\}$ and $h(n) = \{-1, 1\}$ both being zero outside the interval, determine

(a) linear convolution $y(n) = x(n) * h(n)$,

(b) $\tilde{y}(n) = \text{IDFT}\{X(k)\,H(k)\}$. or $x(n) \bigcirc h(n)$

(a)

$$y(0) = x(0)\,h(0) = -1$$

$$y(1) = x(0)\,h(1) + x(1)\,h(0) = -1$$

$y(2) = x(0)h(2) + x(1)h(1) + x(2)h(0) = 2$

thus $\quad y(n) = \{-1, -1, 2\}$ desired linear convolution

Note that the size of this convolution result is $M+N-1=3$ where $M=N=2$ are the size of the input and impulse response sequences. Now take the DFTs of $x(n)$ and $h(n)$

$$X(k) = \sum_{n=0}^{1} x(n) e^{-j\pi kn} \qquad X(k) = x(0) + x(1) e^{-j\pi k}$$

$$X(0) = x(0) + x(1) = 3 \quad , \quad X(1) = x(0) + x(1) e^{-j\pi}$$
$$= 1 + (-1)2 = -1$$

$$\therefore \quad X(k) = \{3, -1\}$$

Similarly

$$H(k) = \{0, -2\}$$

Thus

$$\bar{Y}(k) = H(k)X(k) = \{0, 2\}$$

$$\bar{y}(n) = \frac{1}{2}\sum_{k=0}^{1} \bar{Y}(k) e^{j\pi kn} = \frac{1}{2}\left(\bar{Y}(0) + \bar{Y}(1) e^{j\pi n}\right)$$

which results

$$\bar{y}(n) = \{1, -1\}$$

The discrepancies may be easily seen; by wrapping the sequence $y(n)$ around itself and adding the overlapping first and last terms we obtain $\bar{y}(n)$ i.e.

$$\bar{y}(n) = \{\underbrace{-1, -1}_{+2}\} = \{1, -1\}$$

Note that the last element of $\bar{y}(n)$ is correct. Therefore, $\bar{y}(n)$ can be interpreted as the aliased version of sequence $y(n)$.

To see the process in the time domain

$$\bar{y}(n) = x(n) \,\bigcirc\, h(n)$$

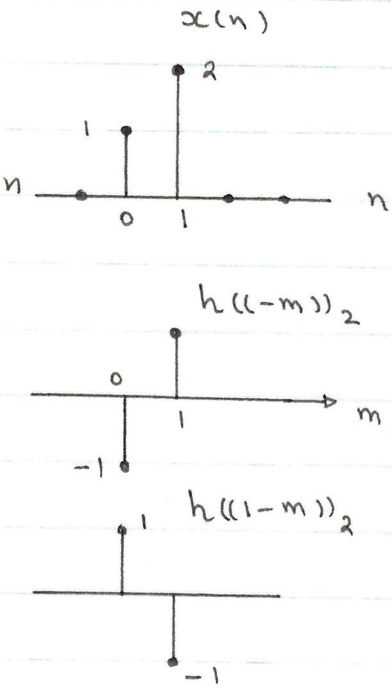
$$= \sum_{m=0}^{1} x((m))_2 \, h((n-m))_2$$

$$= x((0))_2 \, h((n))_2 + x((1))_2 \, h((n-1))_2$$

Thus

$$\bar{y}(0) = x(0)h(0) + x(1)\overbrace{h((-1))_2}^{h(1)} = -1+2 = 1$$

$$\bar{y}(1) = x(0)h(1) + x(1)h(0) = -1$$



Remarks

1- The ~~size~~ of the result of a linear convolution of two finite duration sequences with sizes M and N, is another sequence with size $M+N-1$.

2- The result of circular convolution of two finite duration sequences with sizes M and N, is another finite duration sequence which is the aliased version of the result of linear convolution and is of size M when $M \geq N$.

Linear Convolution Using DFT

Since DFT operations can be performed using fast transform techniques, it is computationally more efficient to implement a linear convolution by computing the DFT's of the sequences.

Let us assume that a finite duration input signal $x(n)$, which contains nonzero values only in region $0 \leq n \leq M-1$ be processed by means of an FIR system with impulse response $h(n)$ that is of size N. If L is the min power of 2 greater than

$$\operatorname{Min}[L = 2^K] \geq M + N - 1$$

the result of circular convolution using DFT gives the linear convolution, if appropriate number of zeros are added to the tails of both sequences to make them of size L. Thus, the steps are

1- Pad both sequences by zeros such that

$$\hat{x}(n) = \begin{cases} x(n) & n = 0, 1, \dots M-1 \\ 0 & n = M, \dots L-1 \end{cases}$$

$$\hat{h}(n) = \begin{cases} h(n) & n = 0, 1 \dots N-1 \\ 0 & n = N, \dots L-1 \end{cases}$$

2- Compute the DFTS of the padded sequences i.e.

$$\hat{X}(K) = DFT_L\{\hat{x}(n)\} \quad \text{and} \quad \hat{H}(K) = DFT_L\{\hat{h}(n)\}$$

and form the product of the Transforms

$$\hat{Y}(K) = \hat{H}(K)\,\hat{X}(K) \qquad K \in [0, L-1]$$

3- Compute the IDFT of $\hat{Y}(K)$ i.e.

$$\hat{y}(K) = IDFT\{\hat{H}(K)\,\hat{X}(K)\}$$

Now let us repeat the previous example using this scheme.

<u>Example</u>

$$x(n) = \{1, 2\}, \quad h(n) = \{-1, 1\}, \quad M = N = 2$$

$$\operatorname{Min}[L = 2^K] \geq M + N - 1 = 3 \Rightarrow L = 4 \quad \text{i.e.}$$

$$\hat{x}(n) = \{1, 2, 0, 0\}, \quad \hat{h}(n) = \{-1, 1, 0, 0\}$$

$$\hat{X}(k) = \sum_{n=0}^{3} \hat{x}(n) e^{-2\pi j \frac{kn}{4}}$$

$$= \hat{x}(0) + \hat{x}(1) e^{-j2\pi \frac{k}{4}}$$

which gives $\hat{X}(k) = \{3, 1-2j, -1, 1+2j\}$

Similarly

$$\hat{H}(k) = \{0, -1-j, -2, -1+j\}$$

Then

$$\hat{Y}(k) = \hat{H}(k)\hat{X}(k) \qquad k = 0, 1, 2, 3$$

$$= \{0, -3+j, 2, -3-j\}$$

Taking IDFT of $\hat{Y}(k)$ yields

$$\hat{y}(n) = \frac{1}{4}\sum_{k=0}^{3} \hat{Y}(k) e^{j2\pi \frac{kn}{4}}$$

$$= \frac{1}{4}\left[\hat{Y}(0) + \hat{Y}(1) e^{j\frac{n\pi}{2}} + \hat{Y}(2) e^{jn\pi} + \hat{Y}(3) e^{j\frac{3\pi n}{2}}\right]$$

which gives

$$\hat{y}(n) = \{-1, -1, 2, \underbrace{X}_{\text{Discard}}\}$$

This is the desired linear convolution result.

<u>Remarks</u>

Although the foregoing procedure is useful in eliminating the wraparound error in the circular convolution it suffers from the following disadvantages.

1- Since L equals to the min. power of 2 greater than $(M+N-1)$, the memory is not used in an efficient way.

2- If $N \ll L$ a waste of time results in the computation of an

L point DFT.

3- often it is required to process a very large input sequence which needs considerable amount of memory. For this kind of appeications the long input sequence is normally sectioned into blocks that are of size comparable to the size of the impulse response sequence and then the filtering is performed on the overlapping sections. These methods are known as "sectioning techniques" and will be discussed later.