Advanced Digital Signal Processing
Lecture 7
Date: 10-2-2004
Scribed by Aditya Thakur

Properties of FT (*contd.*):

Duality property:

Similar to the duality theorem in Boolean algebra wherein $0 \leftrightarrow 1$, AND $\leftrightarrow$ OR.

Here,
Frequency $\leftrightarrow$ time

The IDFT matrix and corresponding DFT matrix

Arrows showing correspondence between the IDFT and DFT rows; angles are negated

$$\begin{pmatrix} \angle 0 & \angle 0 & \angle 0 & \cdots & \angle 0 \\ \angle 0 & \angle {}^{1}/_{N} & \angle {}^{2}/_{N} & \cdots & \angle {}^{N-1}/_{N} \\ \angle 0 & \angle {}^{2}/_{N} & \angle {}^{1}/_{N} & \cdots & \angle {}^{N-1}/_{N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \angle 0 & \angle {}^{N-1}/_{N} & \angle {}^{N-2}/_{N} & \cdots & \angle {}^{1}/_{N} \end{pmatrix}$$

$$\begin{pmatrix} \angle 0 & \angle 0 & \angle 0 & \cdots & \angle 0 \\ \angle 0 & \angle -{}^{1}/_{N} & \angle -{}^{2}/_{N} & \cdots & \angle -{}^{(N-1)}/_{N} \\ \angle 0 & \angle {}^{2}/_{N} & \angle {}^{1}/_{N} & \cdots & \angle {}^{N-1}/_{N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \angle 0 & \angle {}^{1}/_{N} & \angle {}^{2}/_{N} & \cdots & \angle {}^{N-1}/_{N} \end{pmatrix}$$

So if we do $DFT(DFT(x))$, we get a flipped or inverted x at the output. This is because the DFT matrix is orthogonal and symmetric (not conjugate symmetric).

Windowing Theorem: Duality applied to the Convolution theorem:

Point wise multiplication in time $\leftrightarrow$ Convolution in frequency domain

Applications of windowing theorem:

1) Analysis of infinite signals:

Given an infinite signal, we can use a 'suitable' size window to clip it in the time domain, so that it is easier to analyze.

The size of the window depends on the nature of the signal being analyzed:

Slow changing signal $\leftrightarrow$ larger window size

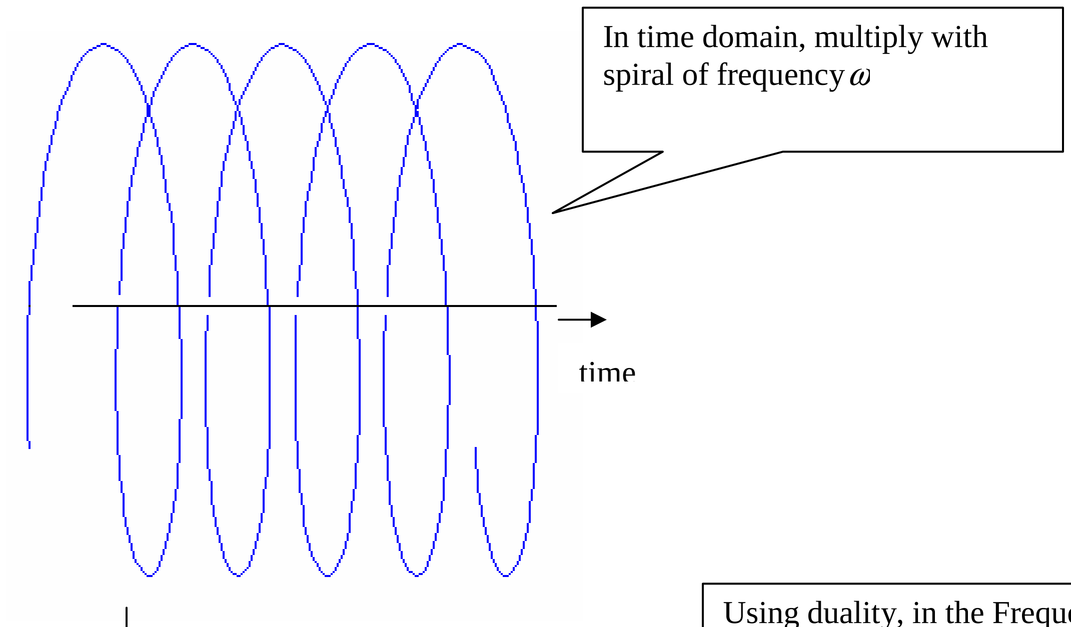
Fast changing signal $\leftrightarrow$ a small window

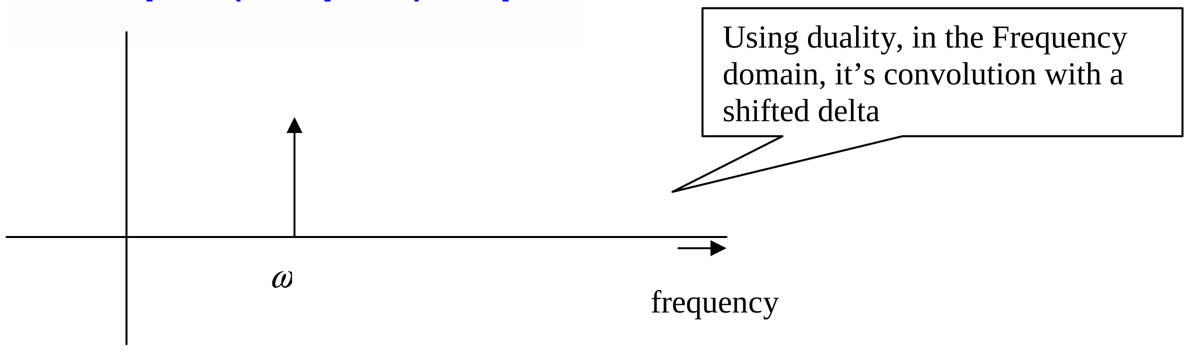
A large window size implies more precise frequency information, while a higher time resolution (small time window) implies worse frequency resolution. This is similar to Heisenberg's Uncertainty Principle, wherein momentum and position, time and energy etc were canonically conjugate to each other.
Another view to explain this phenomenon is by using information theory. Short signals have inherently smaller frequency resolution, and we cannot 'create information'. Of course, using non-linear heuristics it is possible to get better results.

2) Amplitude modulation:

The other application where we point wise multiply in the time domain is in amplitude modulation.

Consider a modulation of a given signal with a single complex spiral having frequency ω. This results in the convolution of the signal with a delta shifted by ω in the frequency domain, which of course results in the frequency shift of the signal from base band to higher band.




Using duality, in the Frequency domain, it’s convolution with a shifted delta
ω
frequency

The above reasoning used the Duality theorem.

One can also think about it directly by decomposing the signal and the modulator into spirals and using the fact that the multiplication of to spirals of frequencies $\omega 1$ and $\omega 2$, results in a spiral of frequency $\omega 1 + \omega 2$ (a frequency shift).

So, the modulation theorem is the dual of the Shift theorem!

Dual of the Derivative theorem:

Multiplication in time domain by a ramp $\leftrightarrow$ differentiation in frequency domain

$$tf(t)dt \qquad \leftrightarrow \qquad \frac{dF}{d\omega}$$

$$\int_{-\infty}^{\infty} tf(t)dt \qquad \leftrightarrow \qquad \frac{dF}{d\omega}$$, at $\omega = 0$ (the dc value)

The LHS represents the expected value of the signal $\mathrm{E}[X]$.
This property is also known as the *First moment property* of the FT.

Similarly,

$$\mathrm{E}[X^2] = \int x^2 f_X(x)dx = \frac{d^2F}{d\omega^2} \text{ at } \omega = 0$$

is called the *Second moment* of the signal.

Together these 2 properties are called the Moment generating properties of the FT, and are used to find the Variance of the signal.

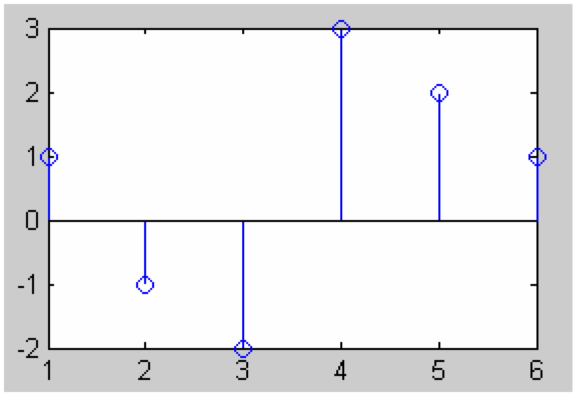
Another interesting point we covered was that when you *add* 2 random variables, their probability mass functions get *convolved*!

Fast Fourier Transform:

2 types:
1) Decimation in time
2) Decimation in frequency

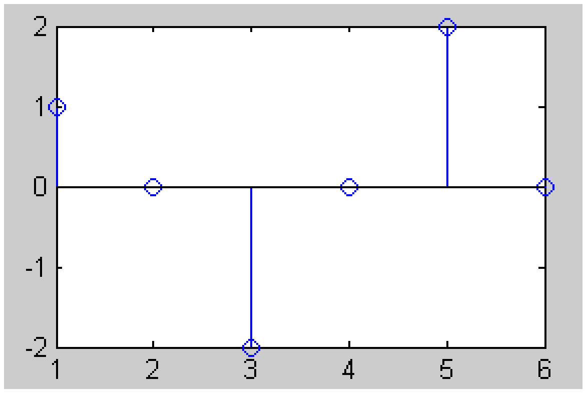
Steps for decimation in time FFT:
(For the following figures, imagine that the x-axis index starts from 0, not 1)

Given a signal 0,

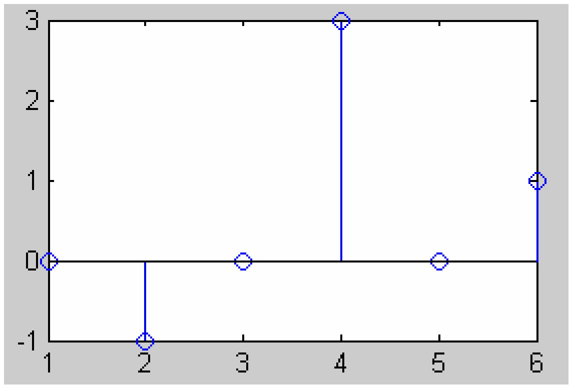


Separate it into even and odd bands 1 & 2 (using linearity),

1: Even band

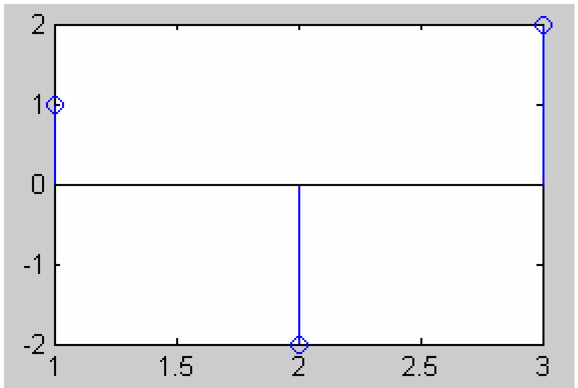


2: Odd band

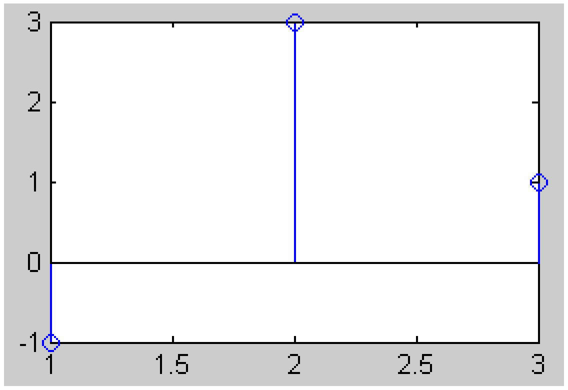


2. Get 1′ & 2′ by down sampling.

1′ *:*



$2'$:



3. Recursively get the DFT of these 2 signals
4. Use stretch theorem (and rotation by w for shifted odd band) to get DFT of 1 & 2
5. Then again use linearity to get DFT of 0

The following recursion represents this process,

$T(n) = T(n/2) + n$
Which has the solution,
$T(n) = \Theta(n \lg n)$

This procedure represents an order improvement over the naive N^2 algorithm. A similar analysis can be carried for decimation in frequency.

Other improvements are possible:

- Sparse matrices are used to get from $1' + 2'$ to 0, since they can be multiplied in linear time.

- Other bases reduce the constant in the $n \lg n$ and are more suitable for use on computers.

- Taking a 4-point instead of 2-point, gives rise to some common sub-expression savings.

- Algorithms such Prime Number decomposition deals with finding DFTs of non-dyadic sequences.

These will be discussed in detail in later lectures.

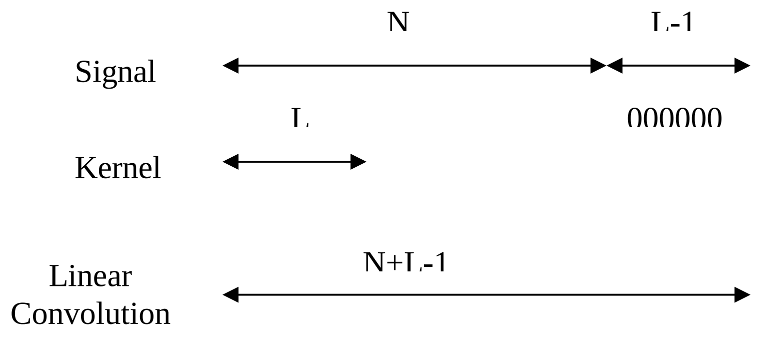
Another interesting point about the FFT is that apart from it being faster it is *more precise*: each coefficient is found after $\lg n$ multiply-adds; as opposed to the n for naïve DFT. This leads to less round-off error.

Interestingly, the best way to minimize round-off error when adding a sequence of numbers is to add the smallest two each time.

So with all our knowledge, we can now do *circular* convolution in $n \lg n$.

What if you wanted to get linear convolution?

Add enough padding to the signal to avoid wraparound distortion.



Can't use such a large FFT because:

- Real-time applications: larger N, more time taken to do the convolution.
- Precision problems: more N, more additions, less precision.

So you would have a case where your kernel is of length 40, your signal is of length 1 million, and you are only allowed (due to the two reasons mentioned above) to take a FFT of 1000!

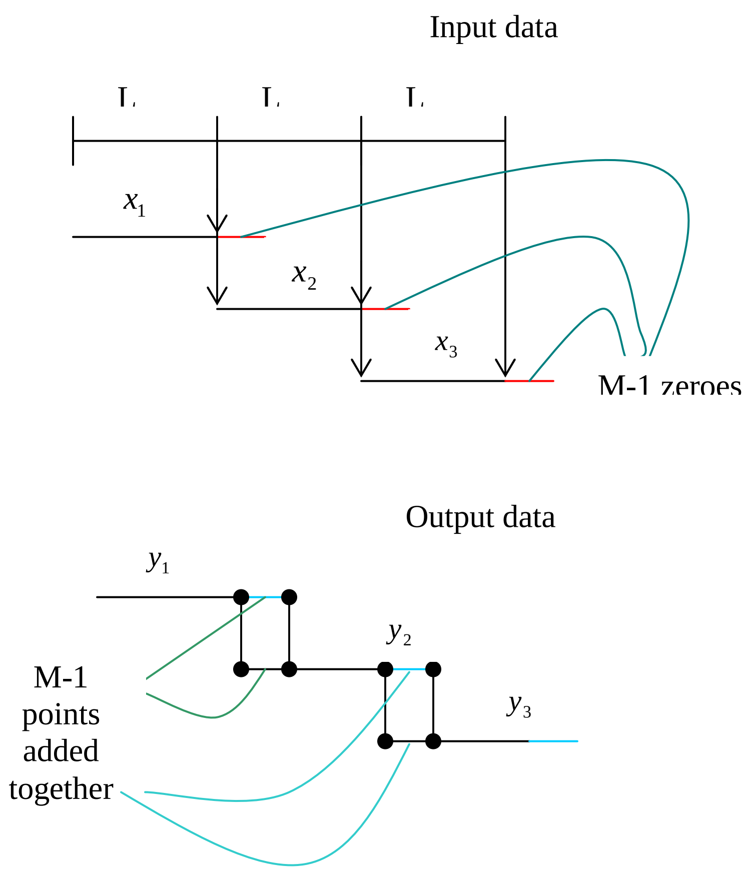
In which case, you use techniques such as *Overlap-add* and *Overlap-save* to do convolution, but don't get the transform.

Overlap-add method:

M	: size of FIR filter kernel
L	: size of the input data block
N=L+M-1	: size of DFT and IDFT

To each data block we append M-1 zeroes and compute N-point DFT. Since each data block is terminated with M-1 zeroes, the last M-1 points from each output block must be overlapped and added to the first M-1 points of the succeeding block.

Overlap-add method

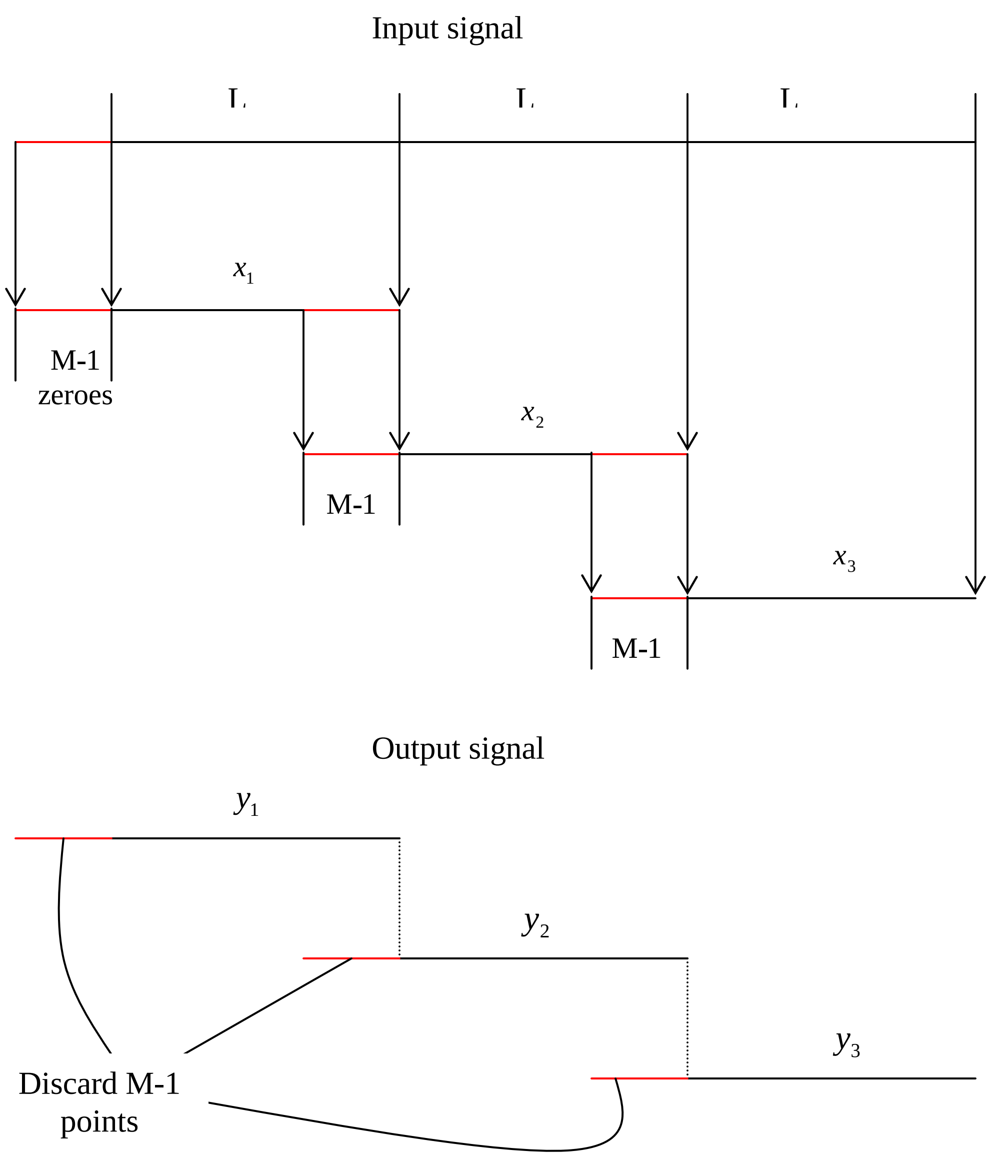


Overlap-save method:

M	: size of FIR filter kernel
N=L+M-1	: size of data blocks
N	: size of DFT and IDFT

Each data block consists of M-1 points from the previous data block followed by L new data points to form a data sequence of N=L+M-1. The kernel length is increased by appending L-1 zeroes and an N-point DFT is computed and stored. The first M-1 points of this are corrupted by aliasing and must be discarded. The last L points are exactly the same as the result from linear convolution.
To avoid loss of data due to aliasing, the last M-1 points of each data record are saved and these points become the first M-1 points of the subsequent record. To begin processing, the first M-1 points are set to zero.

Overlap-save method


Input signal
L
L
L
x_1
M-1
zeroes
x_2
M-1
x_3
M-1
Output signal
y_1
y_2
y_3
Discard M-1
points