

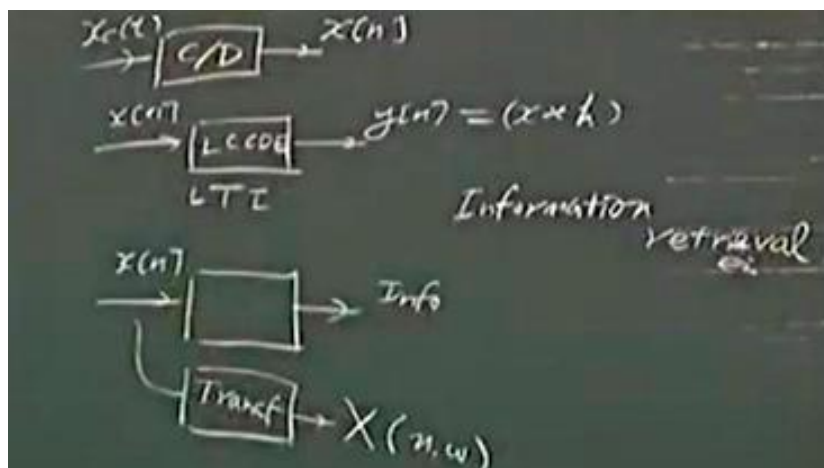


Lecture 10 The Short-time Fourier Transform

Intro. to DSP – Lec 11

Lecture 10: The Short-time Fourier Transform

Prof. Yi-Wen Liu
EE3660: Introduction to DSP
Last updated May 13, 2025



Need for another kind of Fourier transform –
because DTFT is not useful in many applications

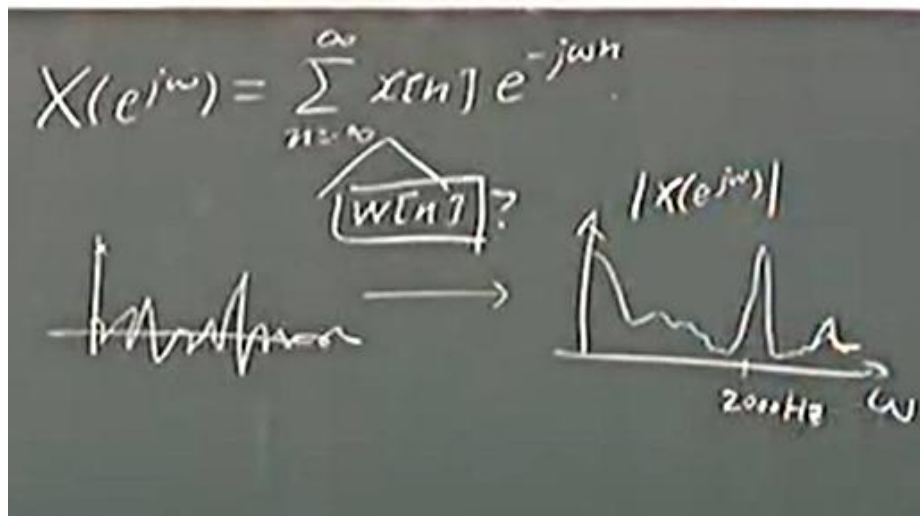
- Query by humming (e.g., soundhound.com)
- Speech coding over internet (e.g. Skype, LINE call, FB messenger call)
- Automatic transcription of music

Notes:

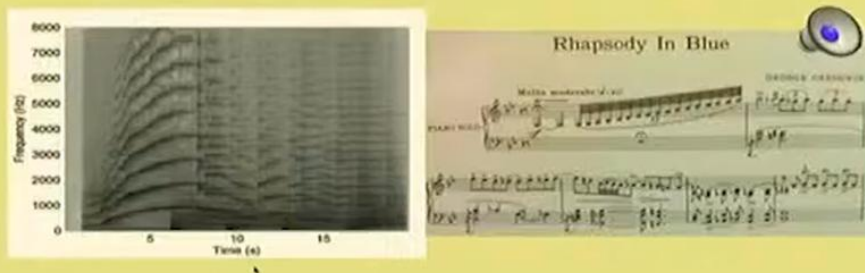
In the real world, sounds are **not stationary** 非平穩的.
However...

DTFT requires **Infinite sum** over time.





Highlight of this lecture: the *spectrogram*



Outline

- The difference between the two
- Relationship between the two
- The fact that the two are not the same
- The difference between the two
- Trade-offs between the two

Warning: Application Error

Message: 1. Application Error

Location: 1. Application Error

Time: 1. Application Error

Stack: 1. Application Error

Warning: Application Error

Message: 1. Application Error

Location: 1. Application Error

Time: 1. Application Error

Stack: 1. Application Error





A: The discrete Fourier transform (DFT)

Summary: $\mathbf{F}^{-1} = \frac{1}{L} \mathbf{F}^*$.

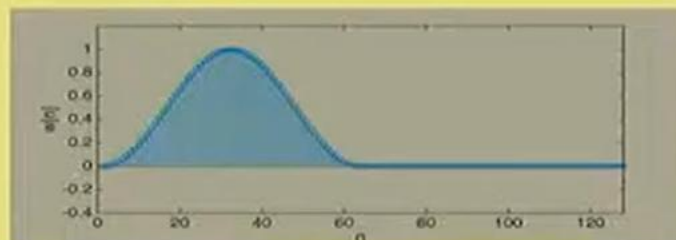
$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & 1 \\ 1 & w & w^2 & \cdots & \cdots & w^{L-1} \\ 1 & w^2 & w^4 & \cdots & \cdots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \cdots & \cdots & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{L-1} \end{bmatrix} \quad w = e^{-j2\pi/L}$$

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{L-1} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & 1 \\ 1 & v & v^2 & \cdots & \cdots & v^{L-1} \\ 1 & v^2 & v^4 & \cdots & \cdots & v^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & v^{L-1} & v^{2(L-1)} & \cdots & \cdots & v^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \vdots \end{bmatrix} \quad v = w^{-1} = e^{j2\pi/L}$$

Short-time Fourier transform: converting a sound to an image

$$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m] \exp(-j\omega n)$$

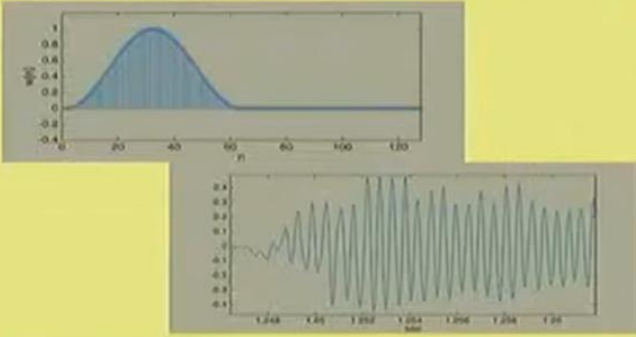
$w[n]$: a **window** with finite number of non-zero points.





Explaining $X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m] \exp(-j\omega n)$

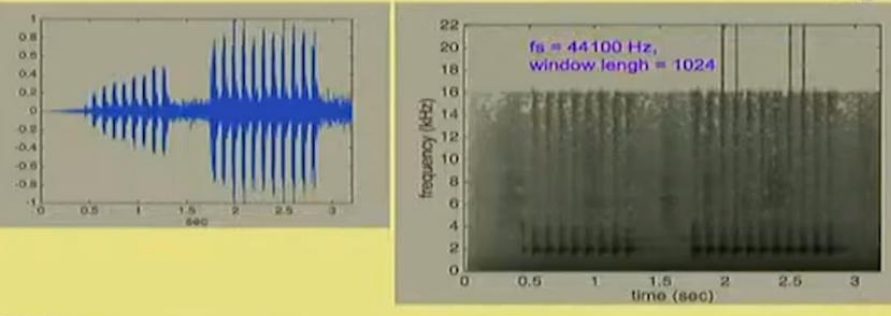
Raised cosine window (also called Hann window)



意義：

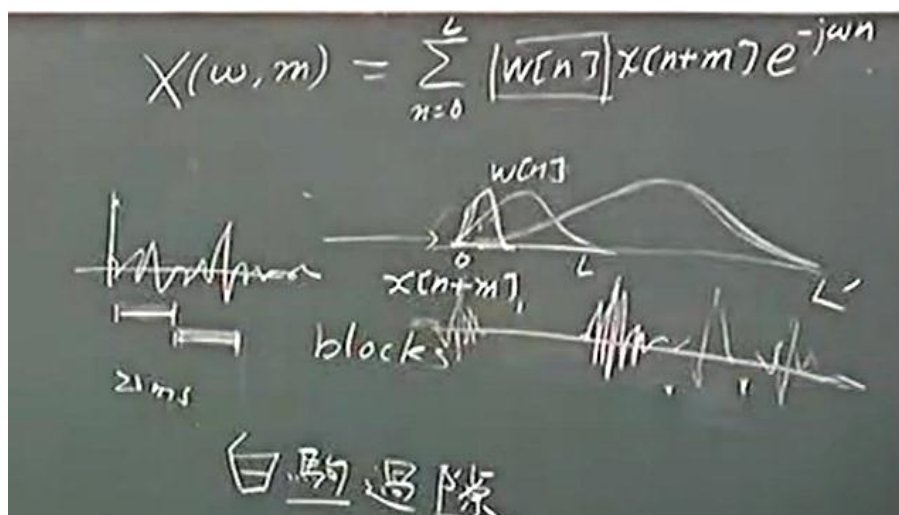
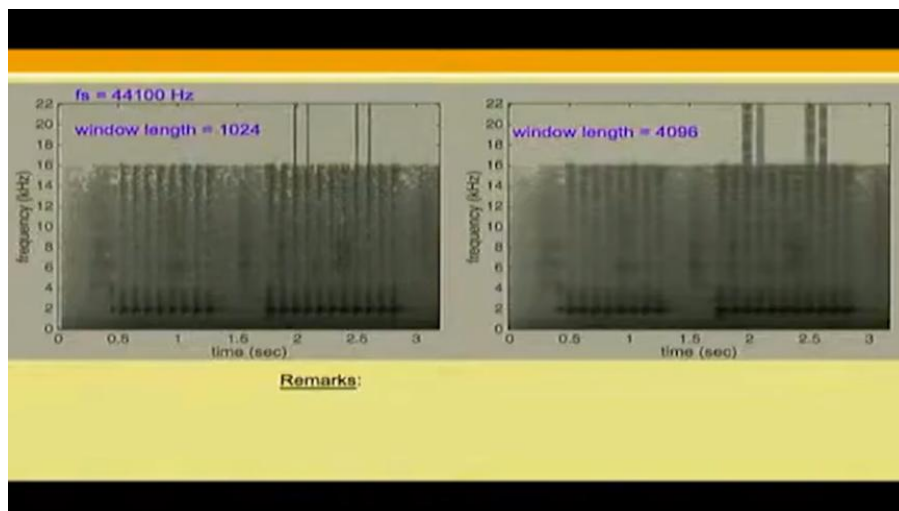
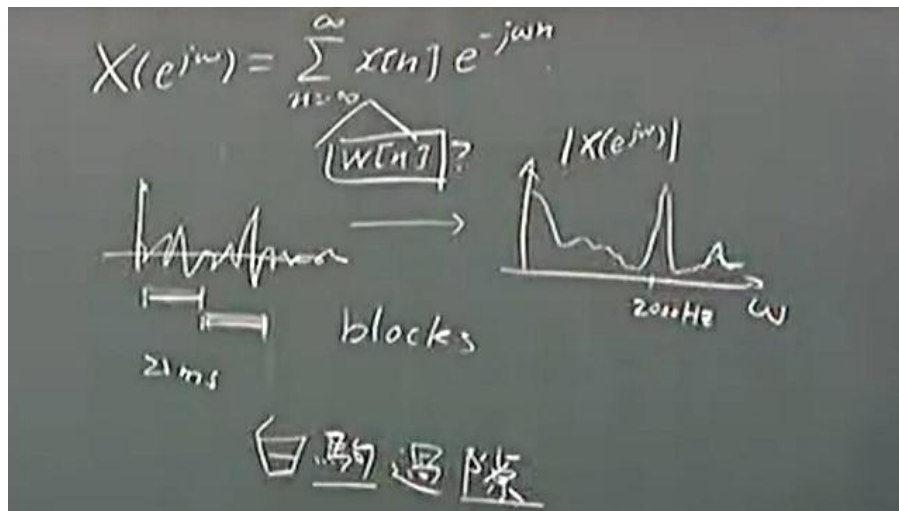
- $x[n+m]$ is like a train going to the left
- $w[n]$ is a fixed window on the ground

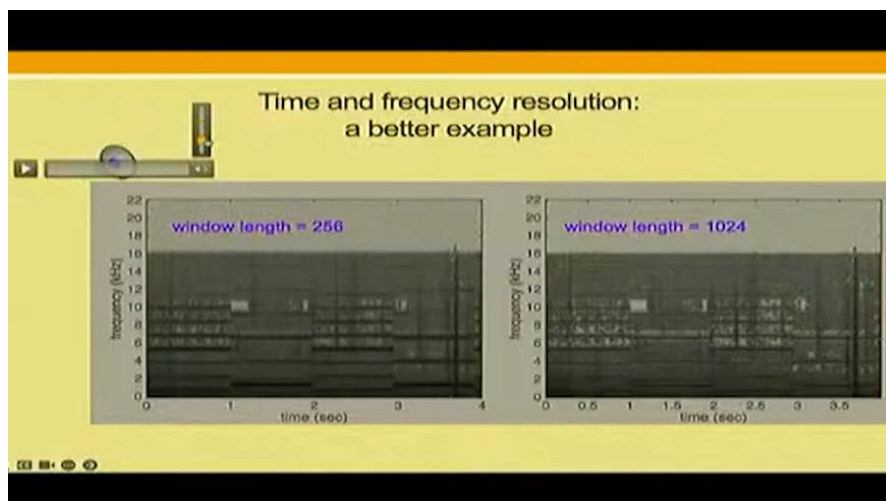
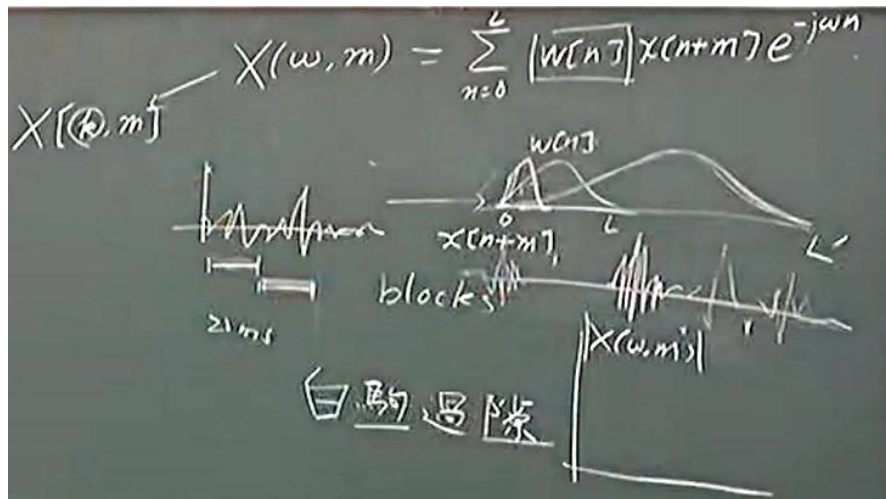
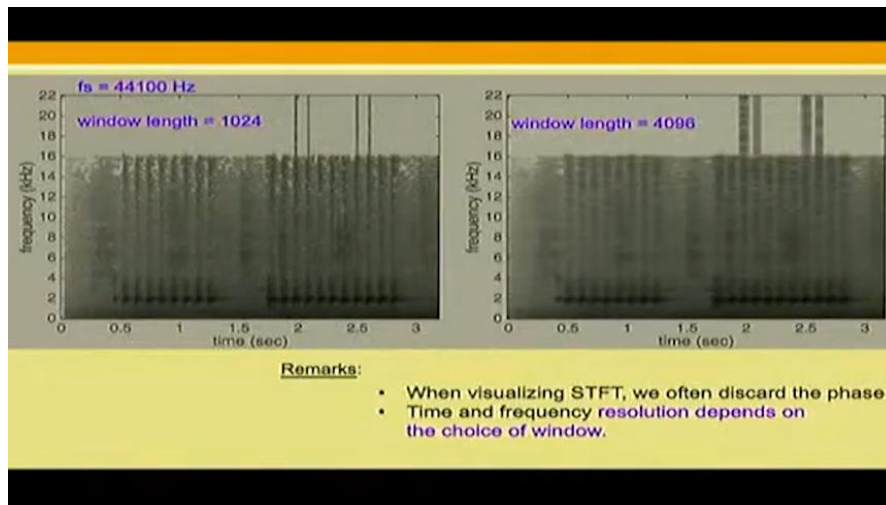
$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m] \exp(-j\omega n)$



fs = 44100 Hz,
window length = 1024



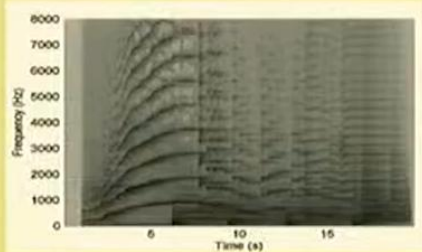






Summary:

sounds are **time-varying spectra** as we hear them.



$$X(\omega, m) = \sum_{n=-\infty}^{\infty} w[n]x[n+m] \exp(-j\omega n)$$

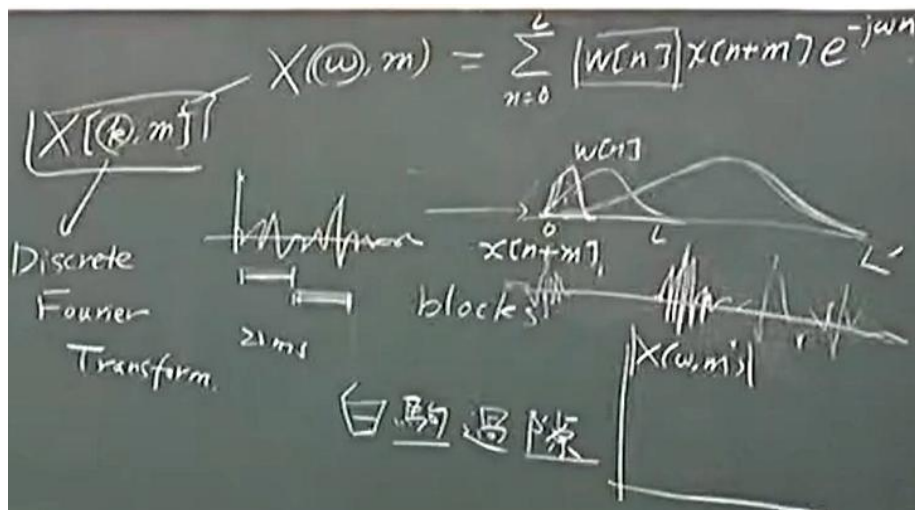
Discrete Fourier Transform (DFT)

Definition: Assume that a block of signal $\mathbf{s}$ contains L points,

$$\mathbf{s} = [s_0, s_1, \dots, s_{L-1}]^T$$

The DFT of $\mathbf{s}$ is defined as a discrete spectrum $\mathbf{S}$ that also contains L points:

$$S_k = \sum_{n=0}^{L-1} s_n \exp\left(-jk \frac{2\pi}{L} n\right), \quad k = 0, 1, 2, \dots, L-1.$$



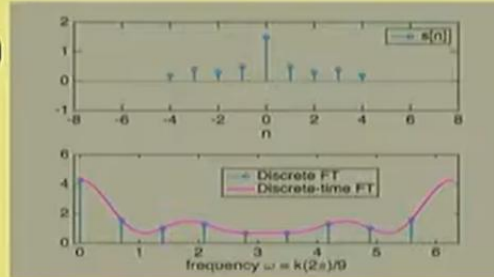
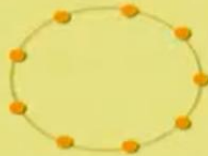


Why is this called *discrete* Fourier transform?

Answer: Because the *discrete* FT is the *discrete-time* FT sampled in frequency.

$$S_k = \sum_{n=0}^{L-1} s_n \exp\left(-jk \frac{2\pi}{L} n\right)$$

$$= S(\omega) \Big|_{\omega = k \frac{2\pi}{L}}$$



Relation between DFT and DTFT

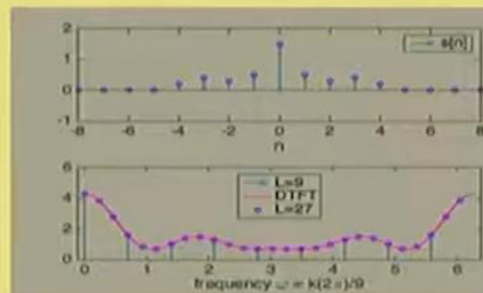
- DFT is DTFT sampled in frequency.

Remarks:

For a finite-length signal:

DTFT is the limit of DFT when L approaches infinity.

* zero-padding



Matrix representation of DFT

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & \cdots & 1 \\ 1 & w & w^2 & \cdots & \cdots & \cdots & w^{L-1} \\ 1 & w^2 & w^4 & \cdots & \cdots & \cdots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \cdots & \cdots & \cdots & w^{(L-1)(L-1)} \end{bmatrix}$$

where $w = e^{-j2\pi/L}$

Note that: $\mathbf{S} \triangleq [S_0, S_1, \dots, S_{L-1}]^T = \mathbf{F}\mathbf{s}$





$$\vec{s} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{L-1} \end{bmatrix} \in \mathbb{R}^L \xrightarrow{\text{DFT}} \vec{\hat{S}} \in \mathbb{C}^L$$

$$\vec{\hat{S}} = [\mathbf{F}] \vec{s}$$

Check:

$$S_k = \sum_{n=0}^{L-1} s_n e^{-jk\frac{2\pi}{L}n}, \quad k=0,1,\dots,L-1$$

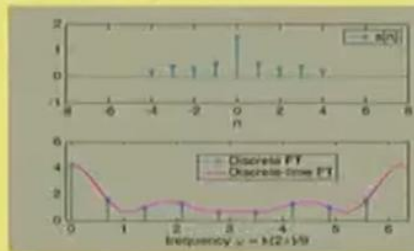
$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Meanings and properties

$$\mathbf{S} \triangleq [S_0, S_1, \dots, S_{L-1}]^T = \mathbf{F}\mathbf{s}$$

DFT is a linear transformation.

Rows of the matrix $\mathbf{F}$ are
orthogonal 正交的.



DFT is a projection onto a new coordinate system

$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^{L-1} \\ 1 & w^2 & w^4 & \dots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \dots & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ \vdots \\ s_{L-1} \end{bmatrix}$$

Notes:

1. Each row represents a basis vector.
2. We calculate their inner products with the signal.
3. Fourier transform is a change of basis from time to frequency.





2B: The Fast Fourier transform (FFT) and its inverse

Regarding the computation load

Recall that

$$\begin{bmatrix} S(0) \\ S(2\pi/L) \\ S(4\pi/L) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & \dots & 1 \\ 1 & w & w^2 & \dots & \dots & w^{L-1} \\ 1 & w^2 & w^4 & \dots & \dots & w^{2(L-1)} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & w^{L-1} & w^{2(L-1)} & \dots & \dots & w^{(L-1)(L-1)} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ \vdots \\ s_{L-1} \end{bmatrix}$$

Therefore, direct computation involves L^2 multiplications.

Is there any way we can reduce the computation load?

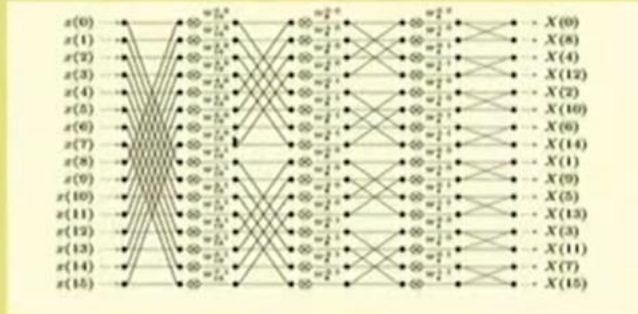
The fast Fourier transform (FFT)

- It turns out that it is possible to reduce the computation from $O(N^2)$ to $O(N \log_2 N)$, and this is generally referred to as FFT.
 - Cooley, J. W. and Tukey, O. W. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Math. Comput.* **19**, 297-301, 1965.
 - Often named one of the 10 most important algorithms in history.
 - Original algorithm is most efficient if $N = 2^p$, e.g. $N = 1024$.





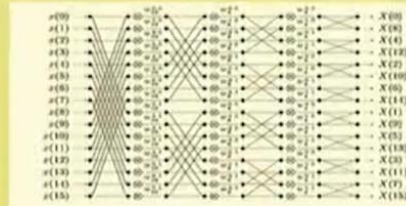
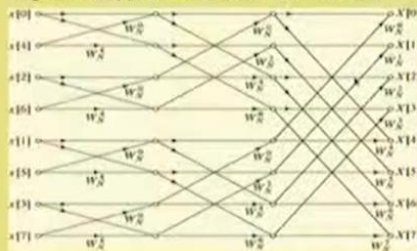
Architecture of the FFT*



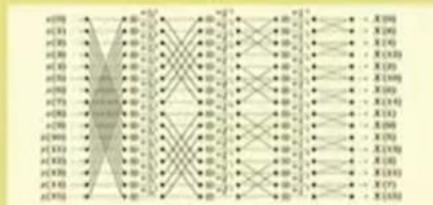
Source code acquired from
<http://www.texample.net/media/tikz/examples/TEX/radix2fft.tex>

Alternative implementations*

Fig. 9-9 of Oppenheim/Schafer DSP textbook

Computation saving is significant when N is large

length of FT	direct matrix: N^2	FFT: $N \log_2 N$	accel.
8	64	24	2.67x
64	4,096	384	10.67x
1024	~1,000,000	10,240	~100x





Handwritten notes on a chalkboard illustrating the Discrete Fourier Transform (DFT) and its application in signal processing.

The main equation shown is:

$$X(\omega, m) = \sum_{n=0}^{L-1} |w[n]| x[n+m] e^{-j\omega n}$$

Below the equation, the term $|w[n]|$ is identified as the **Discrete Fourier Transform**.

A diagram shows a signal $x[n+m]$ being processed in **blocks** of duration $21ms$. The signal is labeled as **fundamental freq.** and **freq.**

Key values and expressions are written:

- $W_{16} = e^{-j\frac{2\pi}{16}}$
- $W_8 = e^{-j\frac{2\pi}{8}}$
- $S(\omega) |_{\omega=k\frac{2\pi}{N}}$
- $\omega = k\frac{2\pi}{N}$
- $|X(\omega, m)|$

At the bottom, a calculation is shown:

$$\boxed{3} \times 16 = 48$$

Below this, the expression $(\log_2 16 - 1) \cdot N$ is written.

