The Fourier Transform

(and more…)

Nimrod Peleg
Nov. 2005

Outline

- Introduce **Fourier series and transforms**
- Introduce Discrete Time Fourier Transforms, (DTFT)
- Introduce Discrete Fourier Transforms (DFT)
- Consider operational complexity of DFT
- Deduce a radix-2 FFT algorithm
- Consider some implementation issues of FFTs with DSPs

The Frequency Domain

- The frequency domain **does not carry any information that is not in the time domain**.
- The power in the frequency domain is that it is simply another way of looking at signal information.
- Any operation or inspection done in one domain is equally applicable to the other domain, except that usually one domain makes a particular operation or inspection *much easier* than in the other domain.
- frequency domain information is extremely important and useful in signal processing.

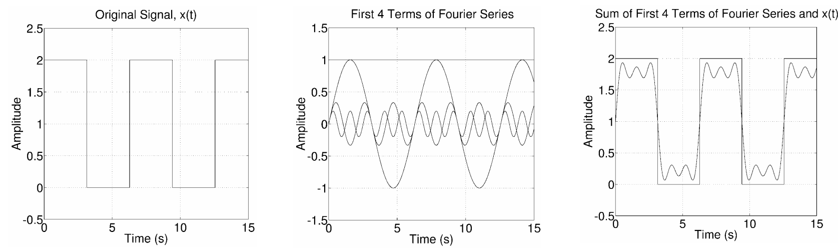
3

3 Basic Representations for FT

- 1. An Exponential Form
- 2. A Combined Trigonometric Form
- A Simple Trigonometric Form

4

The Fourier Series: Exponential Form



Periodic signal expressed as infinite sum of sinusoids.

$$x_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}, \quad \text{where}$$

$$c_k = \frac{1}{T_p} \int_{T_p} x_p(t) e^{-jk\omega_0 t}\, dt$$

Complex Numbers !

- C_k's are frequency domain amplitude and phase representation
- For the given value $x_p(t)$ (a square value), the sum of the first four terms of trigonometric Fourier series are: $\mathbf{x_p(t) \approx 1.0 + \sin(t) + C_2 \sin(3t) + C_3 \sin(5t)}$

5

The Combined Trigonometric Form

- *Periodic signal:* $x_p(t) = x_p(t+T)$ for all t

and cycle time (period) is: $T = \frac{1}{f_0} = \frac{2\pi}{\omega_0}$

f_0 is the fundamental frequency in Hz

ω_0 is the fundamental frequency in radians: $\omega = 2\pi f$

$x_{p(t)}$ can be expressed as an infinite sum of orthogonal functions. When these functions are the cosine and sine, the sum is called the Fourier Series. The frequency of each of the sinusoidal functions in the Fourier series is an *integer multiple of the fundamental frequency*.

Basic frequency + Harmonies

6

Fourier Series Coefficients

- Each individual term of the series, $C_k e^{jk\omega_0 t}$,is the frequency domain representation and is generally complex (frequency and phase), but the ***sum is real.***
- The second common form is the *combined trigonometric form*:

$$x_{p(t)} = C_0 + \sum_{k=1}^{\infty} 2|C_k|\sin(k\omega_0 t + \theta_k)$$

$$\theta_k = \tan^{-1}\frac{\mathrm{Im}(C_k)}{\mathrm{Re}(C_k)}$$

Again: C_k are Complex Numbers !

7

The Trigonometric Form

$$x_{p(t)} = A_0 + \sum_{k=1}^{\infty} A_k Cos(k\omega_0 t) + B_k Sin(k\omega_0 t)$$

$$A_0 = \frac{1}{T_p}\int_{T_p} x_{p(t)}dt = \text{DC} = \text{Average value of } x_{p(t)}$$

$$A_k = \int_{T_p} x_{p(t)} Cos(k\omega_0 t)dt$$

$$B_k = \int_{T_p} x_{p(t)} \mathrm{Sin}(k\omega_0 t)dt$$

Reminder:

$$\mathrm{Cos}\,\theta = \frac{e^{j\theta} + e^{-j\theta}}{2}$$

$$Sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

All three forms are identical and are related using Euler's identity:

$$e^{\pm j\theta} = Cos\theta \pm jSin\theta$$

Thus, the coefficients of the different forms are related by:

$$2C_k = A_k - jB_k \;;\; C_0 = A_0$$

$$\theta_k = \tan^{-1}\frac{B_k}{A_k} = \tan^{-1}\frac{\mathrm{Im}(C_k)}{\mathrm{Re}(C_k)}$$

8

The Fourier Transform 1/3

The Fourier series is only valid for periodic signals.
For non-periodic signals, the *Fourier transform* is used.

Most natural signals are not periodic (speech).
We treat it as a periodic waveform with an *infinite period*.
If we assume that T_P tends towards infinity, then we can produce equations ("model") for non-periodic signals.

If Tp tends towards infinity, then w_0 tends towards 0.
Because of this, we can replace w_0 with dw, and it leads us to:

$$\boxed{1/T_p = f_p = \omega/2\pi} \longrightarrow \lim_{t\to\infty}\frac{1}{T_p} = \frac{d\omega}{2\pi}$$

9

The Fourier Transform 2/3

$$C_{(\omega)} = \frac{d\omega}{2\pi}\int_{-\infty}^{\infty} x_{(t)} e^{-j\omega t}\,dt$$

- Increase T_P = Period Increases : No Repetition: $\frac{1}{T_p} = \frac{\omega}{2\pi} \Rightarrow \frac{d\omega}{2\pi}$
- Discrete frequency variable becomes continuous: $k\omega_0 \Rightarrow \omega$
- Discrete coefficients C_k become continuous: $C_k \Rightarrow C_{(\omega)}$

$$2\pi\frac{C_{(\omega)}}{d\omega} = \int_{-\infty}^{\infty} x_{(t)} e^{-j\omega t}\,dt$$

10

The Fourier Transform 3/3

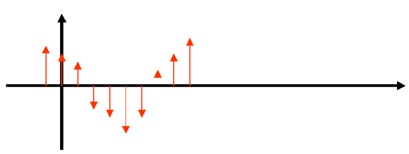
We define: $2\pi \dfrac{C(\omega)}{d\omega} \triangleq X(\omega) = \mathbb{F}\{x(t)\}$

$$\boxed{X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad \Leftrightarrow \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(\omega) e^{j\omega t} d\omega}$$

11

Signal Representation by Delta Function

Instead of a continuous signal we have a “collection of samples”:



This is equivalent to sampling the signal with one Delta Function each time, moving it along X-axis, and summing all the results:

$$x_s(t) = \sum_{-\infty}^{\infty} x(t) \delta(t - nTs)$$

Note that the Delta is “1” only
If its index is zero !

12

Discrete Time Fourier Transform 1/3

- Consider a sampled version, $x_s(t)$, of a continuous signal, x(t) : $x_s(t) = \sum_{-\infty}^{\infty} x(t)\delta(t - nTs)$

Ts is the sample period. We wish to take the Fourier transform of this sampled signal. Using the definition of Fourier transform of $x_{s(t)}$ and some mathematical properties of it we get:

- Replace continuous time t with (nT_s)
- Continuous x(t) becomes discrete x(n)
- Sum rather than integrate all discrete samples

$$x_s(\omega) = \sum_{n=-\infty}^{\infty} x(nTs)e^{-j\omega nTs}$$

13

Discrete Time Fourier Transform 2/3

Fourier Transform $x(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\, dt$ ⇨ $x(\Omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n}$ Discrete Time Fourier Transform

Inverse Fourier Transform $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(\omega)e^{j\omega t}\, d\omega$ ⇨ $x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} x(\Omega)e^{j(n\Omega)}\, d\Omega$ Inverse Discrete Time Fourier Transform

- Limits of integration need not go beyond $\pm\pi$ because the spectrum repeats itself outside $\pm\pi$ (every 2π): $X(\Omega) = X(\Omega + 2\pi)$

- Keep integration because $X(\Omega)$ is continuous: $\Omega = \omega T_s$ means that Ω is periodic every Ts

14

Discrete Time Fourier Transform 3/3

- Now we have a transform from the time domain to the frequency domain that is discrete, but ...

DTFT is not applicable to DSP because it requires an infinite number of samples and the frequency domain representation is a continuous function – impossible to represent exactly in digital hardware.

15

1st result: Nyquist Sampling Rate 1/2

- The Spectrum of a sampled signal is periodic, with 2*Pi Period: $X(\Omega) = X(\Omega + 2\pi)$

Easy to see: $$X(\Omega+2\pi) = \sum_{n=-\infty}^{\infty} x(n) e^{-jn(\Omega+2\pi)} = \sum_{n=-\infty}^{\infty} x(n) e^{-jn\Omega} e^{-j2\pi n}$$

$$= \sum_{n=-\infty}^{\infty} x(n) e^{-jn\Omega} = X(\Omega)$$

$$\boxed{e^{-j2\pi n} = \cos(2\pi n) - j\sin(2\pi n) = 1}$$

16

1^{st} result: Nyquist Sampling Rate 2/2

- For maximum frequency w_H :

$$\Omega|_{\omega=\omega_H} = \pi$$

$$\Omega = \omega Ts$$

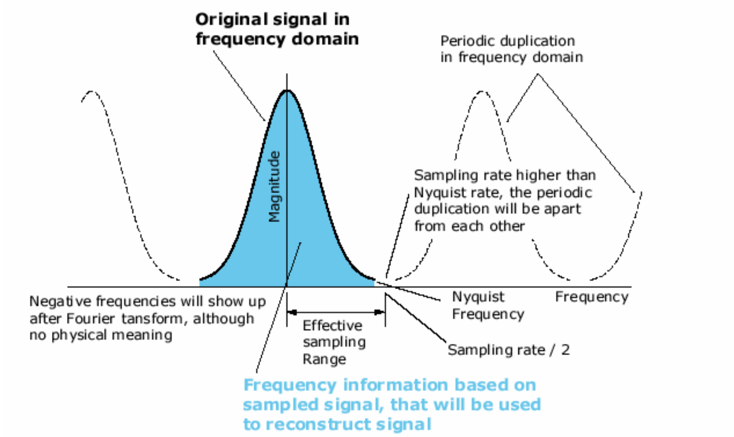
$$\pi = \omega_H Ts$$

$$Ts = \frac{\pi}{\omega_H} \qquad BUT: \quad Ts = \frac{\pi}{\omega_s}$$

$$\Rightarrow \omega_s = \frac{2\pi}{Ts} = 2\omega_H$$

17

Nyquist Conclusion…



Spectrum of Sampled Signal

18

Practical DTFT

- Take only N time domain samples

$$x(\Omega)=\sum_{n=-\infty}^{\infty}x(n)e^{-j\Omega n} \quad \Longrightarrow \quad x(\Omega)=\sum_{n=0}^{N-1}x(n)e^{-j\Omega n}$$

- Sample the frequency domain, i.e. only evaluate x(Ω) at N discrete points. The equal spacing between points is ΔΩ = 2π/N

$$x\left(\frac{2\pi k}{N}\right)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi kn/N} \qquad k=0,1,2,\ldots,N-1$$

The DFT

Since the only variable in $2\pi k/N$ is k, the DTFT is written:

$$x(k)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi kn/N} \qquad k=0,1,2,\ldots,N-1$$

Using the shorthand notation: $W_N=e^{-j2\pi/N}$ (Twiddle Factor)

The result is called Discrete Fourier Transform (DFT):

$$X_N(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn} \quad and \quad x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X_N(k)W_N^{-kn}$$

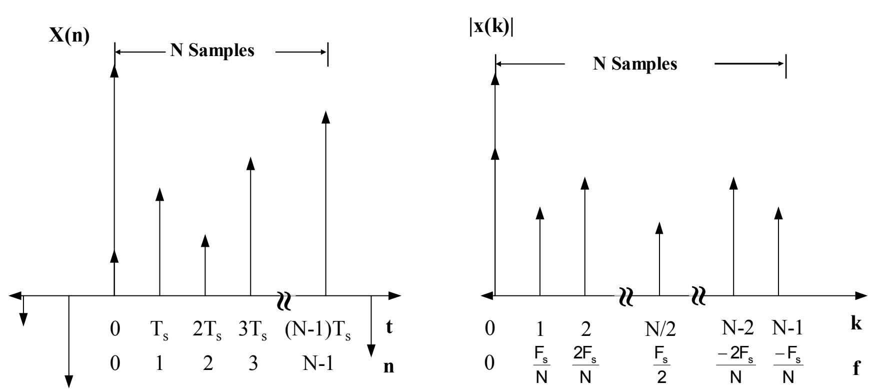
Usage of DFT

- The DFT pair allows us to move between the time and frequency domains while using the DSP.
- The time domain sequence x[n] is discrete and has spacing *T*s, while the frequency domain sequence X[k] is discrete and has spacing 1/*NT* [Hz].

21

DFT Relationships

Time Domain **Frequency Domain**



22

Practical Considerations

- Standard DFT: $$X_N(k) = \sum_{n=0}^{N-1} x_n(k) W_N^{kn} \qquad 0 \le k \ge N-1$$
- An example of an 8 point DFT:
$$X_N(k) = \sum_{n=0}^{7} x_N(k) W_7^{kn} \qquad k = 0,1,2,\ldots,7$$
- Writing this out for each value of n :
$$X_n(k) = x(0)W_7^{k0} + x(1)W_7^{k1} + \ldots\ldots + x(7)W_7^{k7}, k = 0,1,\ldots,7$$
  - Each term such as $\mathbf{x(0)W_7^{k0}}$ requires 8 multiplications
- Total number of (complex !) multiplications required: 8 * 8 = 64
- 1000-point DFT requres $1000^2 = 10^6$ complex multiplications
  And all of these need to be summed….

23

Fast Fourier Transform

Symmetry Property $\quad W_N^{k+N/2} = -W_N^k$

Periodicity Property $\quad W_N^{k+N} = W_N^k$

Splitting the DFT in two (odd and even)
$$X_N(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r).W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1).W_N^{(2r+1)k}$$

or
$$X_N(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r).(W_N^2)^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1).(W_N^2)^{rk}$$

Manipulating the twiddle factor $\quad W_N^2 = e^{j(-\frac{2\pi}{N}2)} = e^{j(-\frac{2\pi}{N/2})} = W_{N/2}$

THE FAST FOURIER TRANSFORM
$$\boxed{X_n(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_{N/2}^{rk}}$$

24

FFT complexity

$$x_N(k) = \underbrace{\sum_{r=0}^{N/2-1} x(2r)W_{N/2}^{rk}}_{(N/2)^2 \text{ multiplications}} + \underbrace{W_N^k}_{N/2 \text{ Multiplications}} \underbrace{\sum_{r=0}^{N/2-1} x(2r+1)W_{N/2}^{rk}}_{(N/2)^2 \text{ multiplications}}$$

- For an 8-point FFT, $4^2 + 4^2 + 4 = 36$ multiplications, saving $64 - 36 = 28$ multiplications
- For 1000 point FFT, $500^2 + 500^2 + 500 = 50{,}500$ multiplications, saving $1{,}000{,}000 - 50{,}500 = 945{,}000$ multiplications

25

Time Decimation

- Splitting the original series into two is called *decimation in time*
- Let us take a short series where N = 8
- Decimate once Called *Radix-2* since we divided by 2

  n = {0, 1, 2, 3, 4, 5, 6, 7}n = { 0, 2, 4, 6 } and { 1, 3, 5, 7 }
- Decimate again

  n = { 0, 4 } { 2, 6 } { 1, 5 } and { 3, 7 }
- The result is a savings of $N^2 - (N/2)\log_2 N$ multiplications:

  1024 point DFT = 1,048,576 multiplications

  1024 point FFT = 5120 multiplication
- Decimation simplifies mathematics but there are more twiddle factors to calculate, and a practical FFT incorporates these extra factors into the algorithm

26

Simple example: 4-Point FFT

- Let us consider an example where N=4: $$X_4(k) = \sum_{0}^{3} x(n) W_4^{kn}$$
- Decimate in time into 2 series:
  n = {0 , 2} and {1, 3}
$$X_4(k) = \sum_{r=0}^{1} x(2r) W_2^{rk} + W_4^k \sum_{r=0}^{1} x(2r+1) W_2^{rk}$$
$$= \{x(0) + x(2)W_2^k\} + W_4^k\{x(1) + x(3)W_2^k\}$$
- We have two twiddle factors.
  Can we relate them?
$$W_N^k = e^{-j\frac{2\pi}{N}k}$$
$$W_2^k = e^{-j\frac{2\pi}{2}k} = e^{-j\frac{2\pi}{4}2k} = W_2^{2k}$$
- Now our FFT becomes:
$$X_{4(k)} = \{x(0) + x(2)W_4^{2k}\} + W_4^k\{x(1) + x(3)W_4^{2k}\}$$

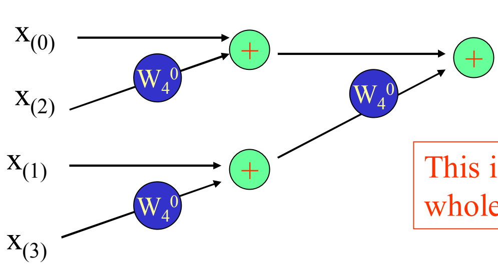
27

4-Point FFT Flow Diagram

The 2 DFT's:
for *k=0,1,2,3*
$$X_{4(k)} = \{x(0) + x(2)W_4^{2k}\} + W_4^k\{x(1) + x(3)W_4^{2k}\}$$

For *k=0* only:
$$X_{4(0)} = \{x(0) + x(2)W_4^{0}\} + W_4^0\{x(1) + x(3)W_4^{0}\}$$

<u>A 'flow-diagram' of it:</u>



This is for only 1/4 of the whole diagram !

28

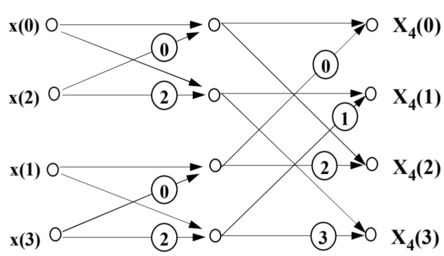
A Complete Diagram

$$X_{4(0)} = \{x(0) + x(2)W_4^0\} + W_4^0\{x(1) + x(3)W_4^0\}$$

$$X_{4(1)} = \{x(0) + x(2)W_4^2\} + W_4^1\{x(1) + x(3)W_4^2\}$$

$$X_{4(2)} = \{x(0) + x(2)W_4^0\} + W_4^2\{x(1) + x(3)W_4^0\}$$

$$X_{4(3)} = \{x(0) + x(2)W_4^2\} + W_4^3\{x(1) + x(3)W_4^2\}$$



Note: $W_N^k = e^{-j\frac{2\pi}{N}k}$

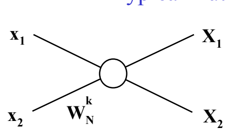
$$W_4^4 = e^{-j\frac{2\pi}{4}4} = 1 = W_4^0$$

$$W_4^6 = e^{-j\frac{2\pi}{4}6} = -1 = W_4^2$$

29

The Butterfly

A Typical Butterfly



$X_1 = x_1 + W_N^k x_2$

$X_2 = x_1 - W_N^k x_2$

Twiddle Conversions

$W_4^0 = 1$

$W_4^1 = -j$

$W_4^2 = -1$

$W_4^3 = j$

4 Point FFT Equations

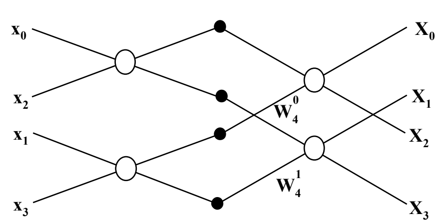
$X_0 = (x_0 + x_2) + W_4^0 \; (x_1 + x_3)$

$X_1 = (x_0 - x_2) + W_4^1 \; (x_1 - x_3)$

$X_2 = (x_0 + x_2) - W_4^0 \; (x_1 + x_3)$

$X_3 = (x_0 - x_2) - W_4^1 \; (x_1 - x_3)$

4 Point FFT Butterfly



30

Summary

- Frequency domain information for a signal is important for processing
- Sinusoids can be represented by phasors
- Fourier series can be used to represent any periodic signal
- Fourier transforms are used to transform signals
  - From time to frequency domain
  - From frequency to time domain
- DFT allows transform operations on sampled signals
- DFT computations can be sped up by splitting the original series into two or more series
- FFT offers considerable savings in computation time
- DSPs can implement FFT efficiently

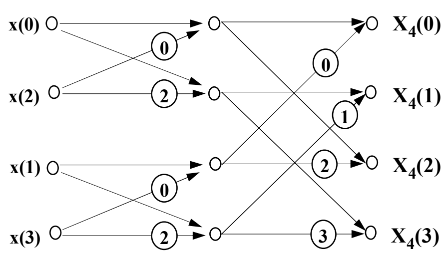
31

Bit-Reversal

- If we look at the inputs to the butterfly FFT, we can see that the inputs are not in the same order as the output.
- To perform an FFT quickly, we need a method of shuffling these input data addresses around to the correct order.
- This can be done either by reversing the order of the bits that make up the address of the data, or by pointer manipulation (bit reversed addition).
- Many DSPs have special addressing modes that allow them to automatically shuffle the data in the background.

32

Bit-Reversal example



- To obtain the output in ascending order the input values must be loaded in the order: {0,2,1,3}
- for 512 or 1024 it is much more complicated...

33

8-point Bit-Reversal

- Consider a 3-bit address (8 possible locations).
- After starting at zero, we add half of the FFT length at each address access with **carrying from left to right (!)**

Start at 0 = 000	=x(0)
000+100 = 100	=x(4)
100+100 = 010	=x(2)
010+100 = 110	=x(6)
110+100 = 001	=x(1)
001+100 = 101	=x(5)
101+100 = 011	=x(3)
011+100 = 111	=x(7)

Note that reversing the order of the address bits gives same result !

34

And what about DCT ???

DCT Type II*:

$$X^{\mathrm{II}}(\mathrm{k}) = \sqrt{\tfrac{2}{N}} \sum_{n=0}^{N-1} \mathrm{b(k)}\, x(n) \cos\left[\frac{\pi(n+\frac{1}{2})k}{N}\right]$$

$$x(n) = \sqrt{\tfrac{2}{N}} \sum_{k=0}^{N-1} \mathrm{b(k)}\, \mathrm{X}^{\mathrm{II}}(\mathrm{k}) \cos\left[\frac{\pi(n+\frac{1}{2})k}{N}\right]$$

$$\mathrm{b(k)} = \begin{cases} 1/\sqrt{2} & \text{if } \mathrm{k}=0 \\ 1 & \text{if } \mathrm{k}=1,\ldots,\mathrm{L}-1 \end{cases}$$

The rest of the math is quite similar…..

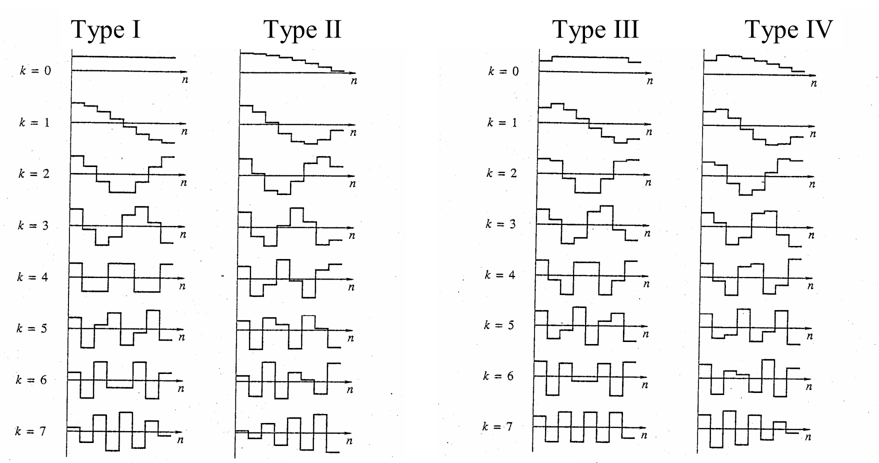
Note: at least 4 types of DCT !!!

* after: A course in Digital Signal Processing, Boaz Porat

35

DCT Type II

Most used for compression:
JPEG, MPEG etc.



DCT Basis Vectors for N=8

36

DCT Features

- Real transformation
- Reversible transformation
- 2D Transformation exists and <u>separable</u>
- Better than the DFT as a "de-correlator"
- Fast algorithm exists (NlogN Complexity)

37

Aliasing, Transforms and More…

- Nyquist's Sampling Theorem is explained here:
  - http://www.cs.cf.ac.uk/Dave/Multimedia/node149.html
- The effect of the above and aliasing is also shown here:
  - http://www.efunda.com/designstandards/sensors/methods/dsp_nyquist.cfm
- And examples to aliasing in images is shown here:
  - http://www.cogs.susx.ac.uk/users/ianw/teach/ms/node19.html
- Java examples of various signal processing:
  - http://www.jhu.edu/~signals/

38