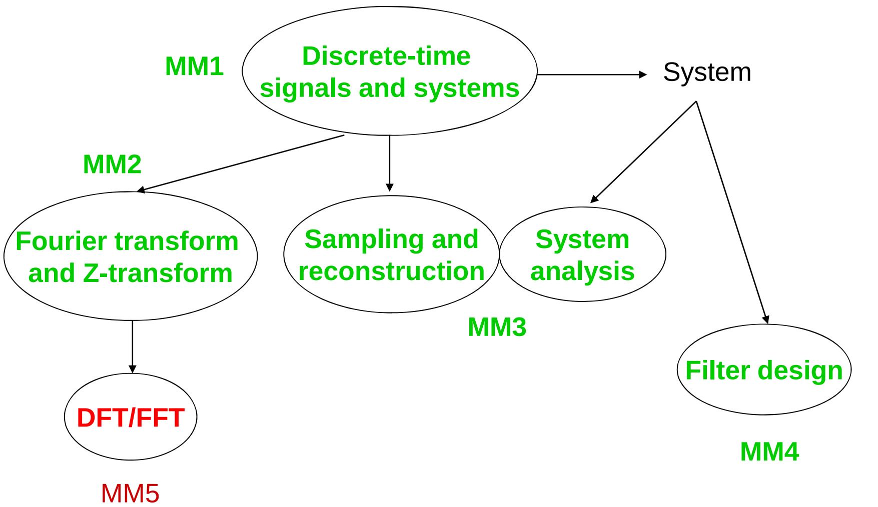
Digital Signal Processing, Fall 2010

Lecture 5: DFT and FFT

Zheng-Hua Tan

Department of Electronic Systems
Aalborg University, Denmark
zt@es.aau.dk

Course at a glance



The discrete-time Fourier transform (DTFT)

- The DTFT is useful for the theoretical analysis of signals and systems.
- But, it has this definition

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

- From the numerical computation viewpoint, the computation of DTFT by computer has several problems:
  - The summation over n is infinite
  - The independent variable w is continuous

  -> DTFT and z-transform are not numerically computable transforms.

A way out

- Goal: find out a numerically computable transform.
- Solution: sample the DTFT in the frequency domain or the z-transform on the unit circle.
- Way to get there:
  - Analyze periodic sequences on the basis that a periodic sequence can always be represented by a linear combination of harmonically related complex exponentials -> Discrete Fourier Series (DFS).
  - Extend the DFS to finite-duration sequences –> Discrete Fourier Transform (DFT), the solution to the two problems!

The discrete Fourier transform (DFT)

- In many cases, only finite duration is of concern
  - The signal itself is finite duration
  - Only a segment is of interest at a time
  - Signal is periodic and thus only finite unique values
- For finite duration sequences, an alternative Fourier representation is DFT
  - The summation over n is finite
  - DFT itself is a sequence, rather than a function of a continuous variable
  - Therefore, DFT is computable and important for the implementation of DSP systems
  - DFT corresponds to samples of the Fourier transform

Part I-A: The discrete Fourier series

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

The discrete Fourier series

- A periodic sequence with period N

$$\tilde{x}[n] = \tilde{x}[n + rN]$$

- Periodic sequence can be represented by a Fourier series, i.e. a sum of complex exponential sequences with frequencies being integer multiples of the fundamental frequency $(2\pi / N)$ associated with the $\tilde{x}[n]$

$$\tilde{x}[n] = \frac{1}{N}\sum_{k} \tilde{X}[k] e^{j(2\pi/N)kn}$$

The frequency of the periodic sequence.

  - Only N unique harmonically related complex exponentials since

$$e^{j(2\pi/N)(k+mN)n} = e^{j(2\pi/N)kn} e^{j2\pi mn} = e^{j(2\pi/N)kn}$$

  - so $\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{j(2\pi/N)kn}$

The Fourier series coefficients

- The coefficients

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{j(2\pi/N)kn}$$

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)kn}$$

- The sequence is periodic with period N

$$\tilde{X}[k+N] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)(k+N)n} = \tilde{X}[k]$$

- For convenience, define $W_N = e^{-j(2\pi/N)}$

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

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

Very similar equations → duality

Part I-B: The discrete Fourier series

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

The Fourier transform of periodic signals

- One conclusion: the DFS coefficients of $\tilde{x}[n]$ are samples of the Fourier transform of the one period of $\tilde{x}[n]$

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

that is

$$\tilde{X}[k] = X(e^{j(2\pi/N)k}) = X(e^{j\omega})|_{\omega=(2\pi/N)k}$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n]e^{-j(2\pi/N)kn}$$

Sampling the Fourier transform

- An aperiodic sequence and its Fourier transform

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \leftrightarrow x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega$$

- Sampling the Fourier transform

$$\tilde{X}[k] = X(e^{j\omega})|_{\omega=(2\pi/N)k} = X(e^{j(2\pi/N)k})$$

  - generates a periodic sequence in k with period N since the Fourier transform is periodic in ω with period 2π

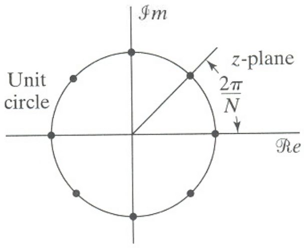


Figure 8.7 Points on the unit circle at which $X(z)$ is sampled to obtain the periodic sequence $\tilde{X}[k]$ ($N = 8$).

RG UNIVERSITY

Example 1

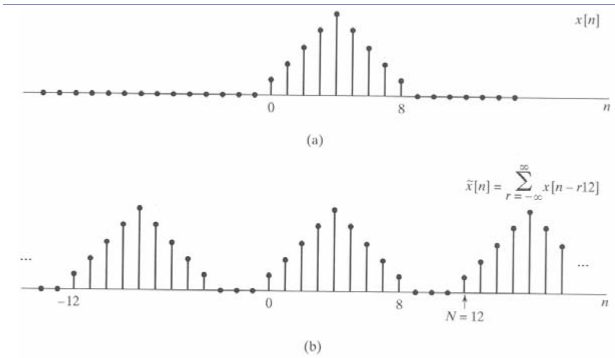


Figure 8.8 (a) Finite-length sequence $x[n]$. (b) Periodic sequence $\tilde{x}[n]$ corresponding to sampling the Fourier transform of $x[n]$ with $N = 12$.

- In this case, the Fourier series coefficients for a periodic sequence are samples of the Fourier transform of one period

AALBORG UNIVERSITY

Example 2

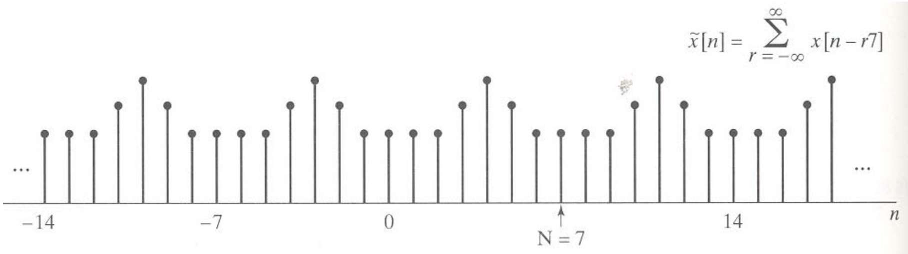


Figure 8.9 Periodic sequence $\tilde{x}[n]$ corresponding to sampling the Fourier transform of $x[n]$ in Figure 8.8(a) with $N = 7$.

- In this case, still the Fourier series coefficients for $\tilde{x}[n]$ are samples of the Fourier transform of $x[n]$. But, one period of $\tilde{x}[n]$ is no longer identical to $x[n]$
- This is just sampling in the frequency domain as compared in the time domain discussed before.

Sampling in the frequency domain

- The relationship between $x[n]$ and one period of $\tilde{x}[n]$ in the undersampled case is considered a form of time domain aliasing.
- Time domain aliasing can be avoided only if $x[n]$ has finite length, just as frequency domain aliasing can be avoided only for signals being bandlimited.
- If $x[n]$ has finite length and we take a sufficient number of equally spaced samples of its Fourier transform (specifically, a number greater than or equal to the length of $x[n]$), then the Fourier transform is recoverable from these samples, equivalently $x[n]$ is recoverable from $\tilde{x}[n]$.

Sampling in the frequency domain

- Recovering $x[n]$

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

i.e. recovering $x[n]$ does not require to know its Fourier transform at all frequencies

- Application: represent finite length sequence by using Fourier series (coefficients) → DFT

$$x[n] \rightarrow \tilde{x}[n] \rightarrow DFS, \tilde{X}[k] \rightarrow \tilde{x}[n] \rightarrow x[n]$$

Sampling the Fourier transform

- Fourier transform $X(j\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega$$

- Discrete-time Fourier transform

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$$

- Discrete Fourier transform

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn}$$

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

Part I-C: The DFT

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

The discrete Fourier transform

- Consider a finite length sequence $x[n]$ of length N samples (if smaller than N, appending zeros)
  - Construct a periodic sequence

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n-rN]$$

Assuming no overlap btw $x[n-rN]$

$$\tilde{x}[n] = x[(n \text{ modulo } N)] = x[((n))_N]$$

  - Recover the finite length sequence

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

- To maintain a duality btw the time and frequency domains, choose one period of $\tilde{X}[k]$ as the DFT

$$X[k] = \begin{cases} \tilde{X}[k], & 0 \le k \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

The DFT

- Periodic sequence and DFS coefficients

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

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

- Since summations are calculated btw 0 and (N-1)

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

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

Generally

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

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

The DFT

- A finite or periodic sequence has only N unique values, $x[n]$ for $0<=n<N$
- Spectrum is completely defined by N distinct frequency samples
- DFT: uniform sampling of DTFT spectrum

The DFT of a rectangular pulse

Example 8.7 pp.561

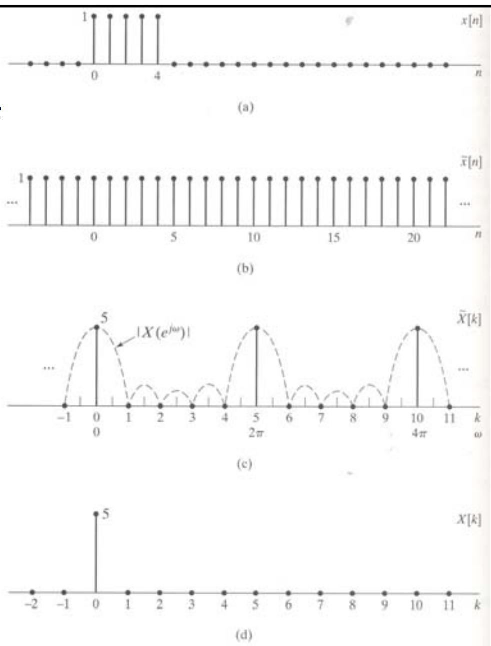


Figure 8.10 Illustration of the DFT. (a) Finite-length sequence $x[n]$. (b) Periodic sequence $\tilde{x}[n]$ formed from $x[n]$ with period $N = 5$. (c) Fourier series coefficients $\tilde{X}[k]$ for $\tilde{x}[n]$. To emphasize that the Fourier series coefficients are samples of the Fourier transform, $|X(e^{j\omega})|$ is also shown. (d) DFT of $x[n]$.

The DFT of a rectangular pul

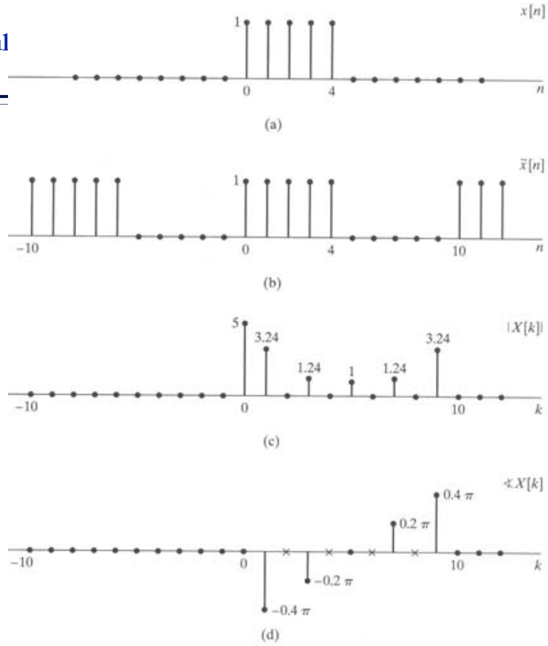


Figure 8.11 Illustration of the DFT. (a) Finite-length sequence $x[n]$. (b) Periodic sequence $\tilde{x}[n]$ formed from $x[n]$ with period $N = 10$. (c) DFT magnitude. (d) DFT phase. (x's indicate indeterminate values.)

Part I-D: Properties of the DFT

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

Properties of the DFT – linearity

Linearity

$$ax_1[n] + bx_2[n] \overset{DFT}{\leftrightarrow} aX_1[k] + bX_2[k]$$

The lengths of sequences and their DFTs are all equal to the maximum of the lengths of $x_1[n]$ and $x_2[n]$

Circular shift of a sequence

- Given

$$x[n] \overset{DFT}{\leftrightarrow} X[k]$$

$$x_1[n] \overset{DFT}{\leftrightarrow} X_1[k] = e^{-j(2\pi k/N)m} X[k]$$

- Then

$$x_1[n] = \begin{cases} \tilde{x}_1[n] = \tilde{x}[n-m] = x[((n-m))_N], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

an example

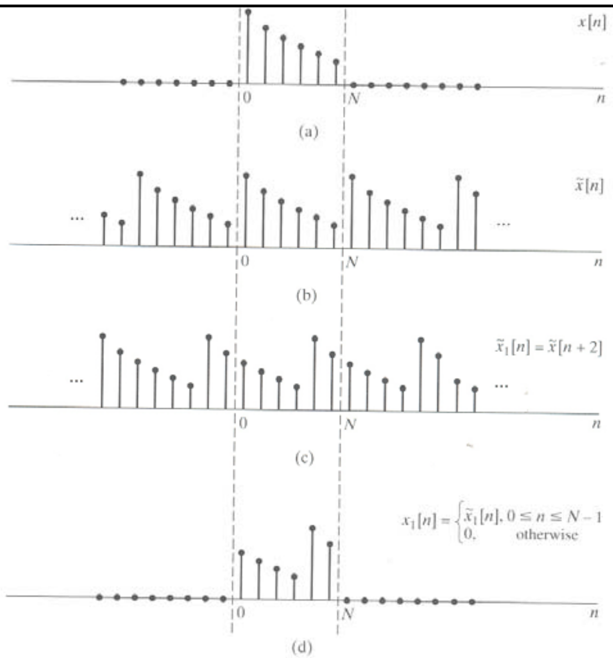


Figure 8.12 Circular shift of a finite-length sequence; i.e., the effect in the time domain of multiplying the DFT of the sequence by a linear phase factor.

Duality

$$x[n] \overset{DFT}{\leftrightarrow} X[k]$$

$$X[n] \overset{DFT}{\leftrightarrow} Nx[((-k))_N], \quad 0 \le k \le N-1$$

Circular convolution

$$x_3[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m], \quad 0 \le n \le N-1$$

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

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

- In linear convolution, one sequence is multiplied by a time –reversed and linearly shifted version of the other. For convolution here, the second sequence is circularly time reversed and circularly shifted. So it is called an N-point circular convolution

$$x_3[n] = x_1[n] \textcircled{N} x_2[n]$$

Circular convolution with a delayed impulse

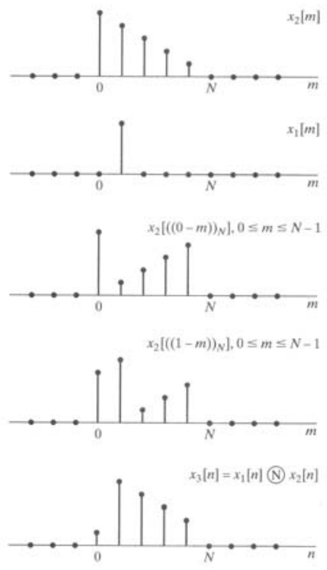


Figure 8.14 Circular convolution of a finite-length sequence $x_2[n]$ with a single delayed impulse, $x_1[n] = \delta[n-1]$.

$$x_1[n] = \delta[n - n_0]$$

$$X_1[k] = W_N^{kn_0}$$

$$X_3[k] = W_N^{kn_0} X_2[k]$$

AALBORG UNIVERSITY

Summary of properties of the DFT

TABLE 8.2

	Finite-Length Sequence (Length N)	N-point DFT (Length N)
1.	$x[n]$	$X[k]$
2.	$x_1[n], x_2[n]$	$X_1[k], X_2[k]$
3.	$ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
4.	$X[n]$	$Nx[((-k))_N]$
5.	$x[((n-m))_N]$	$W_N^{km} X[k]$
6.	$W_N^{-\ell n} x[n]$	$X[((k-\ell))_N]$
7.	$\sum_{m=0}^{N-1} x_1(m) x_2[((n-m))_N]$	$X_1[k]X_2[k]$
8.	$x_1[n]x_2[n]$	$\frac{1}{N}\sum_{\ell=0}^{N-1} X_1(\ell) X_2[((k-\ell))_N]$
9.	$x^*[n]$	$X^*[((-k))_N]$
10.	$x^*[((-n))_N]$	$X^*[k]$
11.	$\mathcal{R}e\{x[n]\}$	$X_{ep}[k] = \frac{1}{2}\{X[((k))_N] + X^*[((-k))_N]\}$
12.	$j\mathcal{I}m\{x[n]\}$	$X_{op}[k] = \frac{1}{2}\{X[((k))_N] - X^*[((-k))_N]\}$
13.	$x_{ep}[n] = \frac{1}{2}\{x[n] + x^*[((-n))_N]\}$	$\mathcal{R}e\{X[k]\}$
14.	$x_{op}[n] = \frac{1}{2}\{x[n] - x^*[((-n))_N]\}$	$j\mathcal{I}m\{X[k]\}$
	Properties 15–17 apply only when $x[n]$ is real.	
15.	Symmetry properties	$\begin{cases} X[k] = X^*[((-k))_N] \\ \mathcal{R}e\{X[k]\} = \mathcal{R}e\{X[((-k))_N]\} \\ \mathcal{I}m\{X[k]\} = -\mathcal{I}m\{X[((-k))_N]\} \\ \lvert X[k]\rvert = \lvert X[((-k))_N]\rvert \\ \sphericalangle\{X[k]\} = -\sphericalangle\{X[((-k))_N]\} \end{cases}$
16.	$x_{ep}[n] = \frac{1}{2}\{x[n] + x[((-n))_N]\}$	$\mathcal{R}e\{X[k]\}$
17.	$x_{op}[n] = \frac{1}{2}\{x[n] - x[((-n))_N]\}$	$j\mathcal{I}m\{X[k]\}$

AALBORG UNIVERSITY

Part I-D: Linear convolution of the DFT

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

Linear convolution using the DFT

- Procedure
  - Compute the N-point DFTs $X_1[k]$ and $X_2[k]$ of two sequences $x_1[n]$ and $x_2[n]$, respectively
  - Compute the product of $X_3[k] = X_1[k]X_2[k]$ for $0 \le k \le N-1$
  - Compute the sequence $x_3[n] = x_1[n] \circledN x_2[n]$ as the inverse DFT of $X_3[k]$
- As we know, the multiplication of DFTs corresponds to a circular convolution of the sequences. To obtain a linear convolution, we must ensure that circular convolution has the effect of linear convolution.

Linear convolution of two finite-length sequences

$$x_3[n] = \sum_{m=-\infty}^{\infty} x_1[m]x_2[n-m]$$

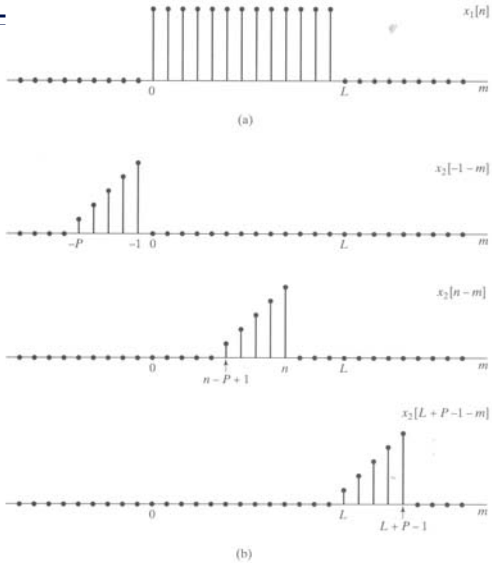


Figure 8.17 Example of linear convolution of two finite-length sequences showing that the result is such that $x_3[n] = 0$ for $n \le -1$ and for $n \ge L + P - 1$. (a) Finite-length sequence $x_1[n]$. (b) $x_2[n - m]$ for several values of n.

Circular convolution as linear convolution with alaising

Fourier transform of $x_3[n]: X_3(e^{j\omega}) = X_1(e^{j\omega})X_2(e^{j\omega})$

Define a DFT: $X_3[k] = X_3(e^{j(2\pi k/N)}), \quad 0 \le k \le N-1$

Also $X_3[k] = X_1(e^{j(2\pi k/N)})X_2(e^{j(2\pi k/N)}), \quad 0 \le k \le N-1$

So, $X_3[k] = X_1[k]X_2[k]$

the inverse DFT of $X_3[k]$:

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n-rN], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$x_{3p}[n] = x_1[n] \, Ⓝ \, x_2[n]$$

The circular convolution corresponding to $X_1[k]X_2[k]$ **is identical to the linear convolution corresponding to** $X_1(e^{j\omega})X_2(e^{j\omega})$ **if the length of DFTs satisfies** $N \ge L + P - 1$

Circular convolution as linear convolution with alaising

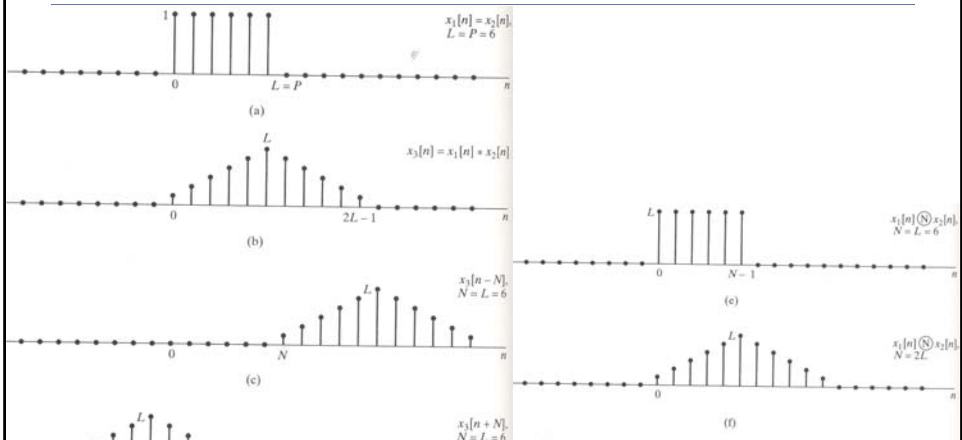


Figure 8.18 Illustration that circular convolution is equivalent to linear convolution followed by aliasing. (a) The sequences $x_1[n]$ and $x_2[n]$ to be convolved. (b) The linear convolution of $x_1[n]$ and $x_2[n]$. (c) $x_3[n-N]$ for $N = 6$. (d) $x_3[n+N]$ for $N = 6$. (e) $x_1[n]$ ⑥ $x_2[n]$, which is equal to the sum of (b), (c), and (d) in the interval $0 \leq n \leq 5$. (f) $x_1[n]$ ⑫ $x_2[n]$.

Part II-A: Direct computation of the DFT

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

Direct computation of the DFT

- The DFT of a finite-length sequence of length *N*

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

- The inverse DFT

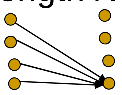
$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}, \qquad n=0,1,\ldots,N-1$$

- Due to the duality, focus on the DFT only.
- Use the number of arithmetic multiplications and additions as a measure of computational complexity.
- Fast Fourier transform (FFT) is a set of algorithms for the efficient and digital computation of the N-point DFT, rather than a new transform.

Direct computation of the DFT

- The DFT of a finite-length sequence of length *N*

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

$x[n]$  $X[k]$

- Direct computation: N^2 complex multiplications and N(N-1) complex additions
  - Compute and store (only over one period)

    $$W_N^k = e^{-j(2\pi/N)k}$$

    $$= \cos(2\pi k/N) + j\sin(2\pi k/N), \quad k=0,1,\ldots,N-1$$

  - Compute the DFT using stored W_N^k and input $x[n]$

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

    W_N^k and $x[n]$ may be complex

Direct computation of the DFT

- For each *k*

$$X[k]=\sum_{n=0}^{N-1}[(\mathrm{Re}\{x[n]\}\,\mathrm{Re}\{W_N^{kn}\}-(\mathrm{Im}\{x[n]\}\,\mathrm{Im}\{W_N^{kn}\})$$

$$+\,j(\mathrm{Re}\{x[n]\}\,\mathrm{Im}\{W_N^{kn}\}+\mathrm{Im}\{x[n]\}\,\mathrm{Re}\{W_N^{kn}\}),\qquad k=0,1,\ldots,N-1$$

- Therefore, for each value of k, the direct computation of X[k] requires 4N real multiplications and (4N-2) real additions.
- The direct computation of the DFT requires $4N^2$ real multiplications and $N(4N-2)$ real additions.
- The efficiency can be improved by exploiting the symmetry and periodicity properties of W_N^{kn}

Symmetry and periodicity of complex exponential

- Complex conjugate symmetry

$$W_N^{k[N-n]}=W_N^{-kn}=(W_N^{kn})^*=\mathrm{Re}\{W_N^{kn}\}-j\,\mathrm{Im}\{W_N^{kn}\}$$

- Periodicity in n and k

$$W_N^{kn}=W_N^{k(n+N)}=W_N^{(k+N)n}$$

- For example

$$\mathrm{Re}\{x[n]\}\,\mathrm{Re}\{W_N^{kn}\}+\mathrm{Re}\{x[N-n]\}\,\mathrm{Re}\{W_N^{k[N-n]}\}$$

$$=(\mathrm{Re}\{x[n]\}+\mathrm{Re}\{x[N-n]\})\,\mathrm{Re}\{W_N^{kn}\}$$

  - The number of multiplications is reduced by a factor of 2.

Part II-B: Decimation-in-time FFT algorithms

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

FFT

- Cooley and Tukey (1965) published an algorithm for the computation of the DFT that is applicable when N is a composite number, i.e., the product of two or more integers. Later, it resulted in a number of highly efficient computational algorithms.
- The entire set of such algorithms are called the fast Fourier transform, FFT.
- FFT decomposes the computation of the DFT of a sequence of length N into successively smaller DFTs.

Decimation-in-time FFT algorithms

- Where
  - decomposition is done by decomposing the sequence into successively smaller subsequences,
  - and both the symmetry and periodicity of complex exponential $W_N^{kn} = e^{-j(2\pi/N)kn}$ are exploited.
- Consider $N = 2^v$ and separate x[n] into two (N/2)-point sequences

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

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

Decimation-in-time FFT algorithms

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

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

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

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

$$= G[k] + W_N^k H[k], \quad k = 0,1,\ldots, N-1$$

(only compute for $k = 0,1,\ldots, N/2-1$) due to the periodicity $(\frac{N}{2})$

Flow graph of the decimation-in-time

- Periodicity is applied, e.g. G[7]=G[3]

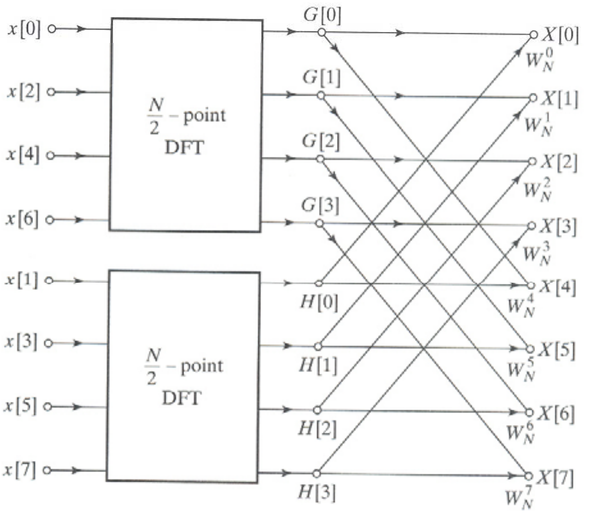


Figure 9.3 Flow graph of the decimation-in-time decomposition of an N-point DFT computation into two $(N/2)$-point DFT computations $(N = 8)$.

Decimation-in-time FFT

- Further break down

$$G[k] = \sum_{r=0}^{(N/2)-1} g[r] W_{N/2}^{rk} = \sum_{l=0}^{(N/4)-1} g[2l] W_{N/2}^{2lk} + \sum_{l=0}^{(N/4)-1} g[2l+1] W_{N/2}^{(2l+1)k}$$

$$= \sum_{l=0}^{(N/4)-1} g[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{(N/4)-1} g[2l+1] W_{N/4}^{lk}$$

$$H[k] = \sum_{l=0}^{(N/4)-1} h[2l] W_{N/4}^{lk} + W_{N/2}^{k} \sum_{l=0}^{(N/4)-1} h[2l+1] W_{N/4}^{lk}$$

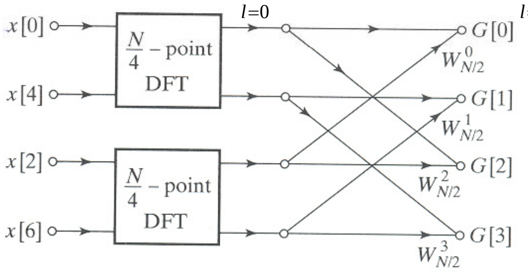


Figure 9.4 Flow graph of the decimation-in-time decomposition of an $(N/2)$-point DFT computation into two $(N/4)$-point DFT computations $(N = 8)$.

Combination of Fig. 9.3 and 9.4

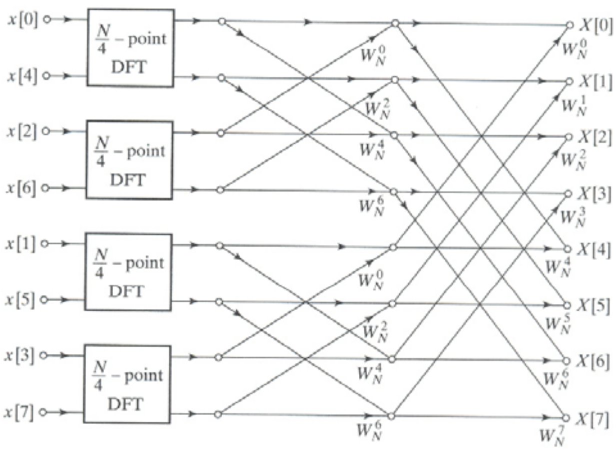


Figure 9.5 Result of substituting the structure of Figure 9.4 into Figure 9.3.

2-point DFT

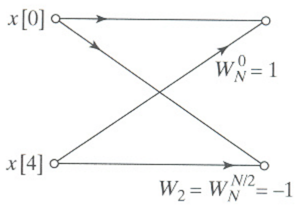
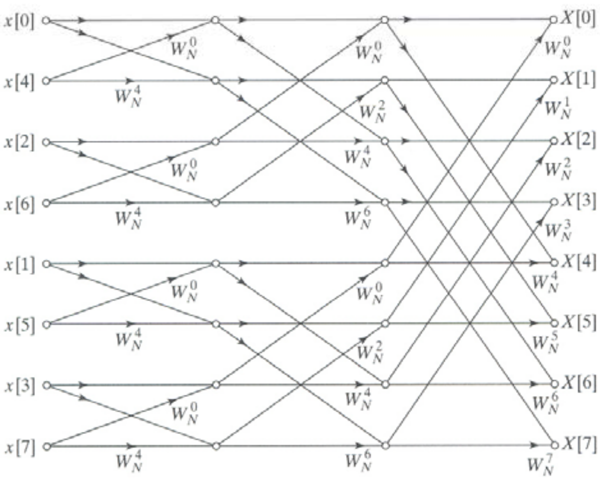


Figure 9.6 Flow graph of a 2-point DFT.

Flow graph



$\log_2 N$ stages and each stage has N complex multiplications and N complex additions .

In total, $N \log_2 N$ complex multiplications and additions

e.g.

$N = 2^{10} = 1024$

$N^2 = 1{,}048{,}576$

$N \log_2 N = 10{,}240$

A reduction of 2 orders!

Figure 9.7 Flow graph of complete decimation-in-time decomposition of an 8-point DFT computation.

Part II-D: Fourier analysis of signals using the DFT

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

Fourier analysis of signals using the DFT

- Finite-duration requirement of DFT → windowing

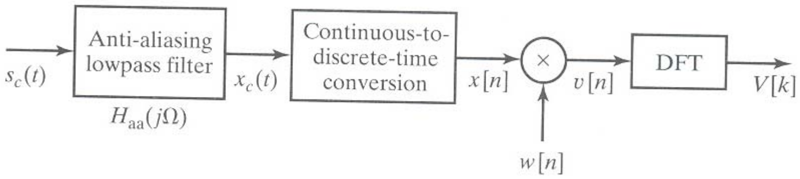
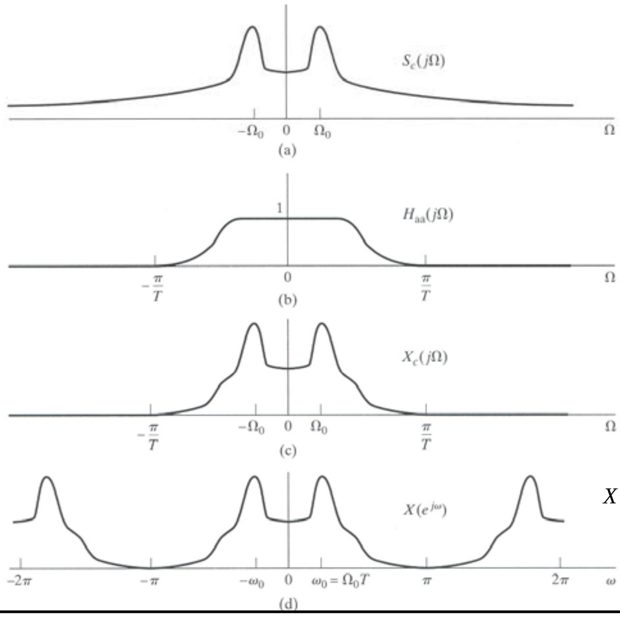


Figure 10.1 Processing steps in the discrete-time Fourier analysis of a continuous-time signal.

Fourier analysis of signals using the DFT



Tapers off but is not band-limited.

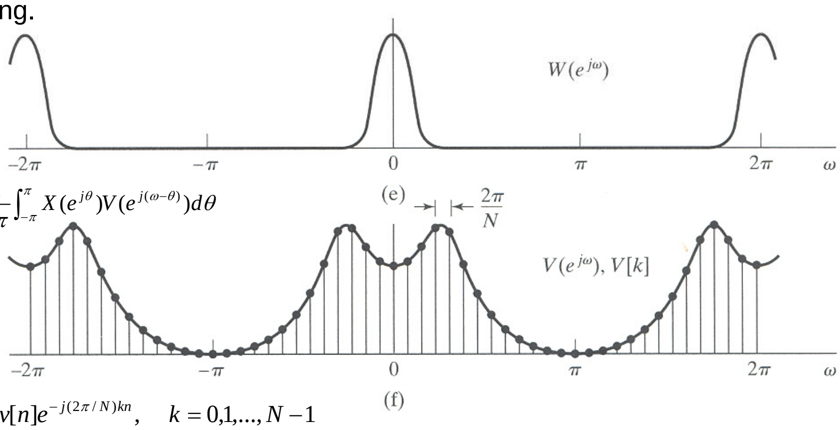
Not ideal.

Low-pass filtered and modified.

$$X(e^{j\omega}) = \frac{1}{T}\sum_{r=-\infty}^{\infty} X_c\left(j\frac{\omega}{T} + j\frac{2\pi r}{T}\right)$$

Fourier analysis of signals using the DFT

Windowing.



$$V(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})V(e^{j(\omega-\theta)})d\theta$$

$$V(k) = \sum_{n=0}^{N-1} v[n]e^{-j(2\pi/N)kn}, \quad k = 0,1,\dots,N-1$$

$$= V(e^{j\omega})|_{\omega=2\pi k/N}$$

Figure 10.2 Illustration of the Fourier transforms of the system of Figure 10.1. (a) Fourier transform of continuous-time input signal. (b) Frequency response of antialiasing filter. (c) Fourier transform of output of antialiasing filter. (d) Fourier transform of sampled signal. (e) Fourier transform of window sequence. (f) Fourier transform of windowed signal segment and frequency samples obtained using DFT samples.

53

Effect of Windowing on Fourier analysis

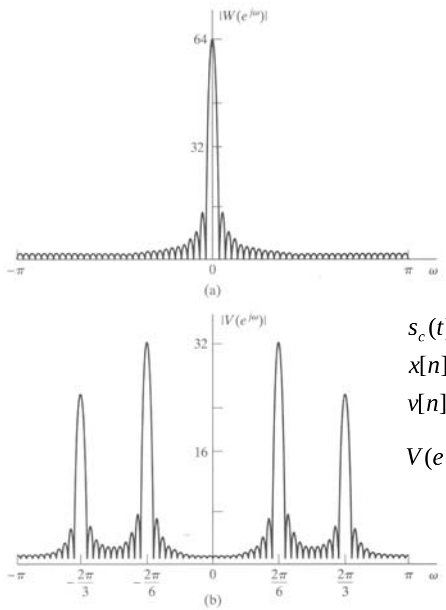


Figure 10.3 Illustration of Fourier analysis of windowed cosines with a rectangular window. (a) Fourier transform of window. (b)–(e) Fourier transform of windowed cosines as $\Omega_1 - \Omega_0$ becomes progressively smaller. (b) $\Omega_0 = (2\pi/6)\times 10^4$, $\Omega_1 = (2\pi/3)\times 10^4$.

A rectangular window of length 64.

$$s_c(t) = A_0\cos(\Omega_0 t + \theta_0) + A_1\cos(\Omega_1 t + \theta_1)$$

$$x[n] = A_0\cos(\omega_0 n + \theta_0) + A_1\cos(\omega_1 n + \theta_1)$$

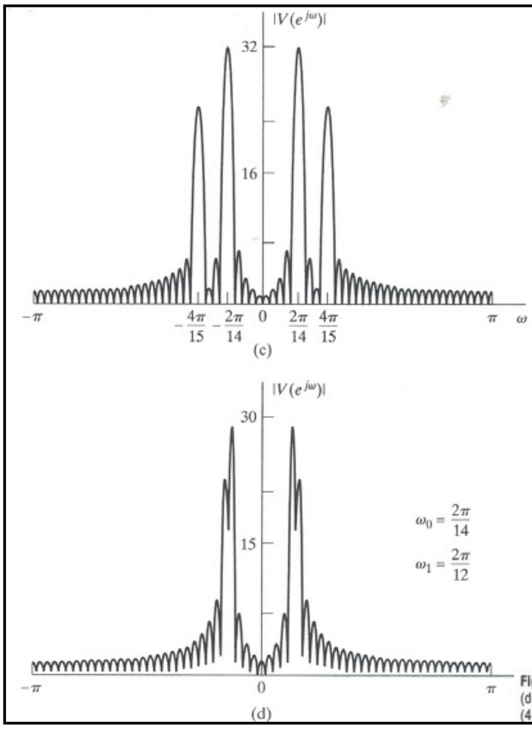
$$v[n] = A_0 w[n]\cos(\omega_0 n + \theta_0) + A_1 w[n]\cos(\omega_1 n + \theta_1)$$

$$V(e^{j\omega}) = \frac{A_0}{2}e^{j\theta_0}W(e^{j(\omega-\omega_0)}) + \frac{A_0}{2}e^{-j\theta_0}W(e^{j(\omega+\omega_0)})$$

$$+\frac{A_1}{2}e^{j\theta_1}W(e^{j(\omega-\omega_1)}) + \frac{A_1}{2}e^{-j\theta_1}W(e^{j(\omega+\omega_1)})$$

AALBORG UNIVERSITY

The DTFT of a sinusoidal signal is a pair of impulses.
Windowing broadens the impulses and reduces the distinction of signals that are close in frequency



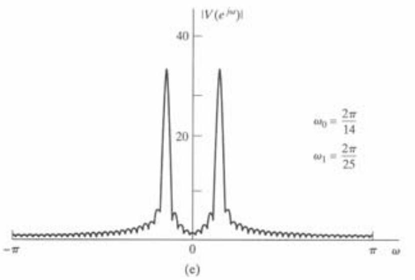
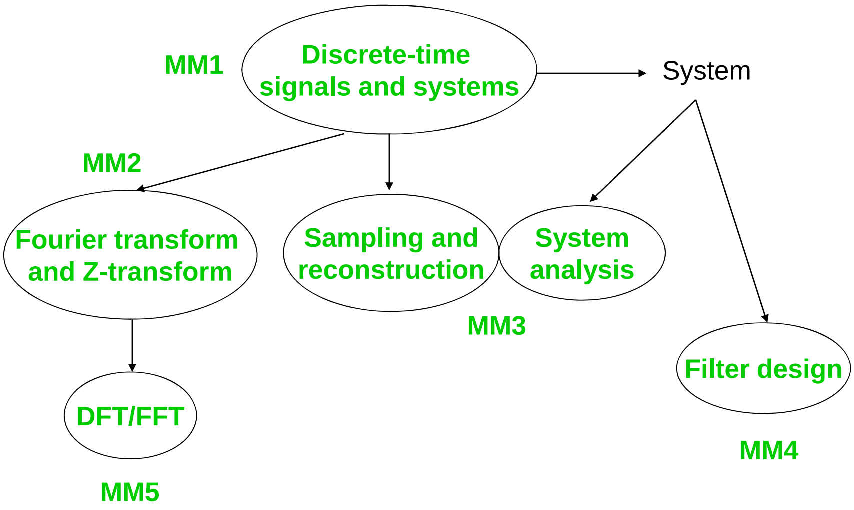


Figure 10.3 *(continued)* (c) $\Omega_0 = (2\pi/14) \times 10^4$, $\Omega_1 = (4\pi/15) \times 10^4$. (d) $\Omega_0 = (2\pi/14) \times 10^4$, $\Omega_1 = (2\pi/12) \times 10^4$. (e) $\Omega_0 = (2\pi/14) \times 10^4$, $\Omega_1 = (4\pi/25) \times 10^4$.

Summary

- DFT
  - The discrete Fourier series
  - Sampling the Fourier transform
  - The discrete Fourier transform
  - Properties of the DFT
  - Linear convolution using the DFT
- FFT
  - Direct computation of the DFT
  - Decimation-in-time FFT algorithms
  - Fourier analysis of signals using the DFT

Course at a glance



The end.

Thanks for your attention!