The Discrete-time Fourier Analysis

Multimedial Signal Processing 1st Module

Politecnico di Milano –

Polo regionale di Como

Particulars

- Eliana Frigerio
  - efrigerio@elet.polimi.it
  - Phone: 02 2399 9653
  - Send to me an email in order to organize conferences and for any question

Summary:

- Ex1: The Discrete-Time Fourier Transform
  - Definition
  - Properties:

Periodicity	Symmetry
Linearity	Sample shifting
Frequency shifting	Conjugation
Folding	Even and Odd
Convolution	Multiplication
Energy – Parseval's Theorem	

Summary:

- Ex2: Frequency Domain Representation of LTI systems
  - Frequency response   Difference equation
- Ex3: Sampling
- Ex4: Reconstruction
  - Ideal D/A converter
  - Zero-order-hold interpolation
  - First-order-hold interpolation
  - Cubic-spline interpolation

Exercise 1 (1/24)

- DTFT: If x(n) is absolutely summable $\sum_{n=-\infty}^{\infty} |x(n)| < \infty$

its DTFT is given by $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$

It is a complex-valued function of the real variable ω, called digital frequency and it is measured in radians.

- The IDTFT is given by: $x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})\, e^{-j\omega n}\, d\omega$

Exercise 1 (2/24)

- OBSERVATIONS:
- DTFT is a continuous function of ω
- ω is a real variable between $-\infty$ and ∞
- We cannot plot properly using Matlab:
  - we have to sample it
  - we have to represent only a part of it
- Using two important properties of the DTFT we can reduce this domain to $[0, \pi]$ for real valued sequences, $[0, 2\pi]$ for any sequence.

Exercise 1 (3/24)

- If x(n) is of infinite duration, Matlab cannot be used to compute $X(e^{j\omega})$. We can use it to evaluate the expression over [0, π].

☐ **EXAMPLE 3.1** Determine the discrete-time Fourier transform of $x(n) = (0.5)^n u(n)$.

Solution The sequence $x(n)$ is absolutely summable; therefore its discrete-time Fourier transform exists.

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n)e^{-j\omega n} = \sum_{0}^{\infty} (0.5)^n e^{-j\omega n}$$

$$= \sum_{0}^{\infty} \left(0.5e^{-j\omega}\right)^n = \frac{1}{1 - 0.5e^{-j\omega}} = \frac{e^{j\omega}}{e^{j\omega} - 0.5}$$ ☐

Exercise 1 (4/24)

EXAMPLE 3.3 Evaluate $X(e^{j\omega})$ in Example 3.1 at 501 equispaced points between $[0, \pi]$ and plot its magnitude, angle, real, and imaginary parts.

- Pseudocode:
- Define the ω axis from 0 to π:

  w = [0:1:500]*pi/500;

  % [0, π] axis divided into 501 points.

- X = exp(j*w) ./ (exp(j*w) - 0.5*ones(1,501));
- plot(w/pi,abs(X))   % we divided the w array by pi before plotting so that the frequeny axes are in the units of π.

Exercise 1 (5/24)

- If x(n) is of finite duration, Matlab can be used to compute $X(e^{j\omega})$ numerically at any frequency ω.

☐ **EXAMPLE 3.2** Determine the discrete-time Fourier transform of the following finite-duration sequence:

$$x(n) = \{1, \underset{\uparrow}{2}, 3, 4, 5\}$$

Solution Using definition (3.1),

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n)e^{-j\omega n} = e^{j\omega} + 2 + 3e^{-j\omega} + 4e^{-j2\omega} + 5e^{-j3\omega}$$

Exercise 1 (6/24)

- **function [X,w]=DTFT(x,n)**
- Define the ω axis from 0 to π divided into 501 points:

  w = [0:1:500]*pi/500;
- X = zeros(1,length(w));
- f_index=1;
- for omega = w
-     t_index=1;
-     for t = n
-         X(f_index)=X(f_index)+ x(t_index)*exp(-1i*(t)*(omega));
-         t_index=t_index+1;
-     end
-     f_index=f_index+1;
- end

Exercise 1 (7/24)

- Pseudocode:
- x= 1:5; n=-1:3;
- [X,w] = DTFT(x,n);

- plot(w/pi,abs(X))

% we divided the w array by pi before plotting so that the frequeny axes are in the units of π.

Exercise 1 (8/24)

If we evaluate DTFT at equispaced frequencies between [0, π], then its computation can be implemented as a matrix-vector multiplication operation:

- assume that $\omega_k \overset{\Delta}{=} \frac{\pi}{M} k \quad k = 0,1,\ldots,M$ which are (M+1) equispaced frequencies between [0, π]
- and x(n) has N samples between $n_1 \le n \le n_N$
- Then: $$X\left(e^{j\omega_k}\right) = \sum_{l=1}^{N} e^{-j\left(\pi/M\right) k n_l} x(n_l)$$

Exercise 1 (9/24)

Rearranging as column vectors: $\overrightarrow{X} = W\vec{x}$

- Then: $W = e^{-j\left(\pi/M\right)k\,n_l} \qquad n_1 \le n \le n_N \qquad k = 0,1,\ldots M$

- rearranging also k and n_l as row vectors

$$W = \left[e^{-j\left(\pi/M\right)\overrightarrow{k^T}\,\vec{n}} \right]_{M+1\mathrm{x}N}$$

Exercise 1 (10/24)

Working with row vectors:

$$\overrightarrow{X^T} = \overrightarrow{x^T} W^T = \overrightarrow{x^T}\left[e^{-j\left(\pi/M\right)\overrightarrow{n^T}\vec{k}} \right]$$

- Pseudocode:
- k = [0:M];
- n = [n1:n2];
- X = x * (exp(-j*pi/M)) .^ (n'*k);

EXAMPLE 3.4 Numerically compute the discrete-time Fourier transform of the sequence $x(n)$ given in Example 3.2 at 501 equispaced frequencies between $[0, \pi]$.

Exercise 1 (11/24)

- PROPERTIES:
- Periodicity: The DTFT is periodic in ω with period 2π

$$X(e^{j\omega}) = X(e^{j(\omega+2\pi)})$$

- Implication: we need only a period for analysis and not the whole domain.

Exercise 1 (12/24)

EXAMPLE 3.5 Let $x(n) = (0.9 \exp(j\pi/3))^n$, $0 \le n \le 10$. Determine $X(e^{j\omega})$ and investigate its periodicity.

x(n) is a complex-valued signal, so it is periodic with period [0, 2π]

We will evaluate and plot X at 401 frequencies over two periods to observe its periodicity:

- n = 0:10; x = (0.9*exp(j*pi/3)).^n;
- k = -200:200; w = (pi/100)*k;

% divide by 100 because we are observing X from -2pi till 2pi

- X = x * (exp(-j*pi/100)) .^ (n'*k);

Exercise 1 (13/24)

- Symmetry: for real-valued x(n), DTFT is conjugate symmetric $$X(e^{-j\omega}) = X^*(e^{j\omega})$$

- or
$$\mathrm{Re}\{X(e^{-j\omega})\} = \mathrm{Re}\{X(e^{j\omega})\} \qquad \textit{even symmetry}$$
$$\mathrm{Im}\{X(e^{-j\omega})\} = -\mathrm{Im}\{X(e^{j\omega})\} \qquad \textit{odd symmetry}$$
$$\left|X(e^{-j\omega})\right| = \left|X(e^{j\omega})\right| \qquad \textit{even symmetry}$$
$$\angle X(e^{-j\omega}) = \angle X(e^{j\omega}) \qquad \textit{odd symmetry}$$

Exercise 1 (14/24)

EXAMPLE 3.6 Let $x(n) = 2^n$, $-10 \leq n \leq 10$. Investigate the conjugate-symmetry property of its discrete-time Fourier transform.

x(n) is a real-valued signal

We will evaluate and plot X at 401 frequencies over two periods to observe its symmetry property:

- n = -5:5; x = (-0.9).^n;
- n = 0:10; x = (0.9*exp(j*pi/3)).^n;
- k = -200:200; w = (pi/100)*k;

% divide by 100 because we are observing X from -2pi till 2pi

- X = x * (exp(-j*pi/100)) .^ (n'*k);

Exercise 1 (15/24)

- Linearity:

$$F\left\{\alpha x_1(n)+\beta x_2(n)\right\}=\alpha F\left\{x_1(n)\right\}+\beta F\left\{x_2(n)\right\}$$

EXAMPLE 3.7 In this example we will verify the linearity property (3.5) using real-valued finite-duration sequences. Let $x_1(n)$ and $x_2(n)$ be two random sequences uniformly distributed between $[0,1]$ over $0 \le n \le 10$. Then we can use our numerical discrete-time Fourier transform procedure as follows.

Exercise 1 (16/24)

- Time shifting: a shift in the time domain corresponds to the phase shifting

$$F\{x(n-k)\} = X\left(e^{j\omega}\right)e^{-j\omega k}$$

EXAMPLE 3.8 Let $x(n)$ be a random sequence uniformly distributed between $[0,1]$ over $0 \leq n \leq 10$ and let $y(n) = x(n-2)$. Then we can verify the sample shift property (3.6) as follows.

Exercise 1 (17/24)

- <u>Frequency shifting</u>: multiplication by a complex exponential correspond to a shift in the frequency domain:

$$F\left\{x(n)\,e^{j\omega_o n}\right\} = X\left(e^{j(\omega-\omega_o)}\right)$$

EXAMPLE 3.9 To verify the frequency shift property (3.7), we will use the graphical approach. Let

$$x(n) = \cos(\pi n/2), \quad 0 \leq n \leq 100 \qquad \text{and} \qquad y(n) = e^{j\pi n/4}x(n)$$

Exercise 1 (18/24)

- Conjugation: conjugation in the time domain corresponds to the folding and conjugation in the frequency domain

$$F\left\{x^{*}(n)\right\} = X^{*}\left(e^{-j\omega}\right)$$

EXAMPLE 3.10 To verify the conjugation property (3.8), let $x(n)$ be a complex-valued random sequence over $-5 \le n \le 10$ with real and imaginary parts uniformly distributed between $[0, 1]$. The MATLAB verification is as follows.

Exercise 1 (19/24)

- Folding: folding in the time domain corresponds to the folding in the frequency domain:

$$F\left\{x(-n)\right\} = X\left(e^{-j\omega}\right)$$

EXAMPLE 3.11 To verify the folding property (3.9), let $x(n)$ be a random sequence over $-5 \le n \le 10$ uniformly distributed between $[0, 1]$. The MATLAB verification is as follows.

Exercise 1 (20/24)

- <u>Even and odd properties:</u> any kind of sequence can be decompose in even and odd parts:

$$x(n) = x_e(n) + x_o(n)$$

- with

$$F\{x_e(n)\} = \mathrm{Re}\{X(e^{j\omega})\}$$
$$F\{x_o(n)\} = j\,\mathrm{Im}\{X(e^{j\omega})\}$$

- Implication: if the sequence x(n) is real and even, then X is also real and even.

Exercise 1 (21/24)

EXAMPLE 3.12 In this problem we verify the symmetry property (3.10) of real signals. Let

$$x(n) = \sin(\pi n/2), \quad -5 \le n \le 10$$

Then using the **evenodd** function developed in Chapter 2, we can compute the even and odd parts of $x(n)$ and then evaluate their discrete-time Fourier transforms. We will provide the numerical as well as graphical verification.

Exercise 1 (22/24)

- Convolution:

$$F\left\{x_1(n) * x_2(n)\right\} = F\left\{x_1(n)\ \right\}F\left\{x_2(n)\right\} = X_1\left(e^{j\omega}\right)X_2\left(e^{j\omega}\right)$$

- see Exercise1 for a Matlab example

Exercise 1 (23/24)

- Multiplication:

$$F\{x_1(n)\cdot x_2(n)\} = F\{x_1(n)\}\otimes F\{x_2(n)\} =$$

$$= \frac{1}{2\pi}\int X_1\left(e^{j\theta}\right)X_2\left(e^{j(\omega-\theta)}\right)d\theta$$

- it corresponds to a periodic convolution (we will se in Chapter 5).

Exercise 1 (24/24)

- Energy: Parseval's theorem

$$E_x = \sum_{-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| X\left(e^{j\omega}\right) \right|^2 d\omega$$

- Definition: Energy density spectrum:

$$\Phi_x(\omega) = \frac{\left| X\left(e^{j\omega}\right) \right|^2}{\pi}$$

Summary:

- Ex2: Frequency Domain Representation of LTI systems
  - Frequency response
  - Difference equation

Exercise 2 (1/13)

- Frequency Response: The DTFT of an impulse response is called the Frequency Response/ Transfer Function of a LTI system:

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-j\omega n}$$

- Because of: $x(n) \longrightarrow \boxed{h(n)} \longrightarrow y(n) = x(n) * h(n)$
- Thanks to the convolution property, for any arbitrary absolute summable sequence: $\sum_{n=-\infty}^{\infty} |x(n)| < \infty$

$$Y\left(e^{j\omega}\right) = X\left(e^{j\omega}\right) H\left(e^{j\omega}\right)$$

Exercise 2 (2/13)

☐ **EXAMPLE 3.13** Determine the frequency response $H(e^{j\omega})$ of a system characterized by $h(n) = (0.9)^n u(n)$. Plot the magnitude and the phase responses.

Solution Using (3.16),

$$H(e^{j\omega}) = \sum_{-\infty}^{\infty} h(n)e^{-j\omega n} = \sum_{0}^{\infty} (0.9)^n e^{-j\omega n}$$

$$= \sum_{0}^{\infty} (0.9e^{-j\omega})^n = \frac{1}{1 - 0.9e^{-j\omega}}$$

Hence

$$\left|H(e^{j\omega})\right| = \sqrt{\frac{1}{(1 - 0.9\cos\omega)^2 + (0.9\sin\omega)^2}} = \frac{1}{\sqrt{1.81 - 1.8\cos\omega}}$$

and

$$\angle H(e^{j\omega}) = -\arctan\left[\frac{0.9\sin\omega}{1 - 0.9\cos\omega}\right]$$

Exercise 2 (3/13)

- Difference Equation: A LTI system can be described by a linear constant coefficient difference equation:

$$\sum_{k=0}^{N} a_k \, y(n-k) = \sum_{m=0}^{M} b_m \, x(n-m) \quad \forall n$$

if $a_N \neq 0$, then the difference equation is of order N.

Exercise 2 (4/13)

- Difference Equation:

$$y(n) = \underbrace{\underbrace{\sum_{m=0}^{M} b_m\, x(n-m)}_{\text{FIR (MA)}} - \underbrace{\sum_{k=1}^{N} a_k\, y(n-k)}_{\text{IIR (AR)}}}_{\text{IIR (ARMA)}}$$

AutoRegressive Moving Average

Exercise 2 (5/13)

- Goal: computation of the impulse response and the output of a digital filter in accordance with the difference equation
- Matlab provides the function: "y=filter(num,den,x)" that computes the output y of the filter defined by the coefficients "b" and "a" when the input is "x". N.B. length(y)=length(x)

Exercise 2 (6/13)

EXAMPLE 2.9 Given the following difference equation

$$y(n) - y(n-1) + 0.9y(n-2) = x(n); \quad \forall n$$

a. Calculate and plot the impulse response $h(n)$ at $n = -20, \ldots, 100$.
b. Calculate and plot the unit step response $s(n)$ at $n = -20, \ldots, 100$.

- HINT: Pay attention to the fact that Matlab indexes start from 1 and not 0 as in Difference Equation

$$y(n) = b_0\, x(n-0) - \sum_{k=1}^{2} a_k\, y(n-k)$$

$$a(1)\,y(n) = -a(2)\; y(n-1) - a(3)\, y(n-2) + b(1)\, x(n)$$

Exercise 2 (7/13)

$$1 * y(n) = 1 * y(n-1) - 0.9 * y(n-2) + 1 * x(n)$$

- a=[1,-1,0.9];
- b=1;
- x=impseq(0,-20,120);   n=[-20:120];

- h=filter(b,a,x);

- N.B.: nh=n

Exercise 2 (8/13)

- Unit step response:

- a=[1,-1,0.9];
- b=1;
- x=stepseq(0,-20,120);   n=[-20:120];

- s=filter(b,a,x);

Exercise 2 (9/13)

- The transfer function of a LTI can be defined as:

$$H\left(e^{j\omega}\right)=\frac{\sum_{m=0}^{M} b_m\, e^{-j\omega m}}{1+\sum_{k=1}^{N} a_k\, e^{-j\omega k}}$$

- Where the difference equation is:

$$\sum_{k=0}^{N} a_k\, y(n-k)=\sum_{m=0}^{M} b_m\, x(n-m) \quad \forall n$$

Exercise 2 (10/13)

EXAMPLE 3.15 An LTI system is specified by the difference equation

$$y(n) = 0.8y(n-1) + x(n)$$

a. Determine $H(e^{j\omega})$.

$$H\left(e^{j\omega}\right) = \frac{1}{1-0.8e^{-j\omega}}$$

Exercise 2 (11/13)

- GOAL: compute the transfer function of a IIR filter:

$$H\left(e^{j\omega}\right)=\frac{\sum_{m=0}^{M} b_m\, e^{-j\omega m}}{1+\sum_{k=1}^{N} a_k\, e^{-j\omega k}}$$

- If we evaluate H at K+1 equispaced frequencies over [0, π], then:

$$H\left(e^{j\omega_k}\right)=\frac{\sum_{m=0}^{M} b_m\, e^{-j\omega_k m}}{1+\sum_{l=1}^{N} a_l\, e^{-j\omega_k l}} \qquad k=0,1,\ldots,K$$

Exercise 2 (12/13)

$$H\left(e^{j\omega_k}\right) = \frac{\sum_{m=0}^{M} b_m \, e^{-j\omega_k m}}{1 + \sum_{l=1}^{N} a_l \, e^{-j\omega_k l}} \qquad k = 0,1,\dots,K$$

- Defining the vectors: $\{b_m\}$ $\{a_l\}$ (with $a_0 = 1$)
  $\{m\} = [0, \dots, M]$
  $\{l\} = [0, \dots, N]$
  $\{\omega_k\}$ with $k = 0, \dots, K$
- Then numerator and denominator become:

$$\vec{b}\exp(-j\overrightarrow{m}^T\vec{\omega}) \qquad \vec{a}\exp(-j\vec{l}^T\vec{\omega})$$

Exercise 2 (13/13)

EXAMPLE 3.16 A 3rd-order lowpass filter is described by the difference equation

$$y(n) = 0.0181x(n) + 0.0543x(n-1) + 0.0543x(n-2) + 0.0181x(n-3) + 1.76y(n-1) - 1.1829y(n-2) + 0.2781y(n-3)$$

Plot the magnitude and the phase response of this filter and verify that it is a lowpass filter.

- b = [0.0181, 0.0543, 0.0543, 0.0181];
- a = [1.0000, -1.7600, 1.1829, -0.2781];
- m = 0:length(b)-1; l = 0:length(a)-1;
- K = 500; k = 0:1:K; w = pi*k/K;

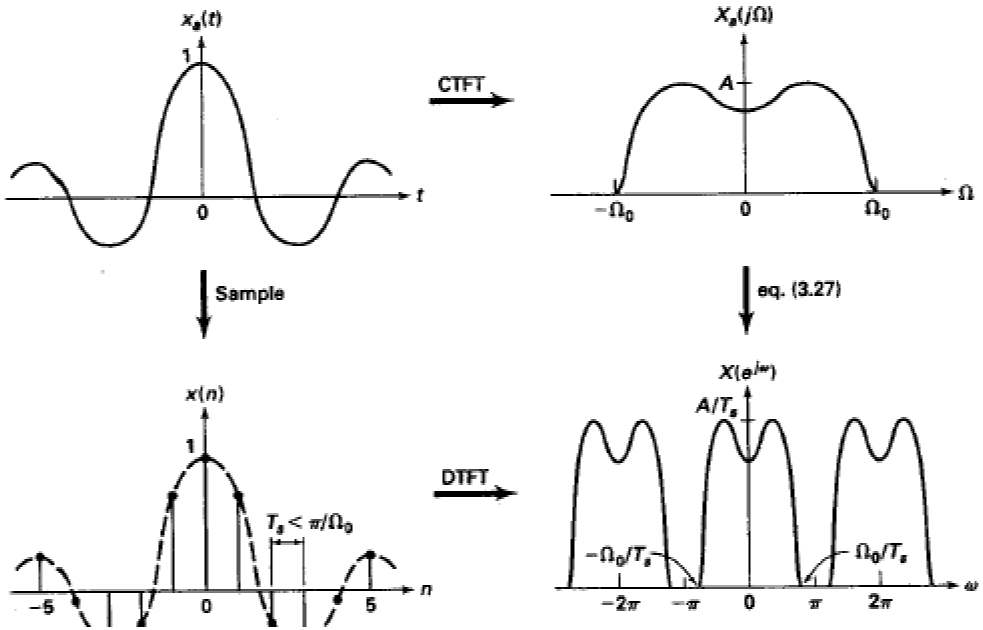
  % [0, pi] axis divided into 501 points.
- num = b * exp(-j*m'*w); den = a * exp(-j*l'*w);
- H = num ./ den;

Summary:

- Ex3: Sampling

Exercise 3 (1/9)

- We know that if: $x(n) = x_a(nT_s)$
- Then: $$X\left(e^{j\omega}\right) = \frac{1}{T_s}\sum_{l=-\infty}^{\infty} X_a\left[j\left(\frac{\omega}{T_s} - \frac{2\pi}{T_s}l\right)\right]$$



Exercise 3 (2/9)

- Sampling Principle: A band-limited signal $x_a(t)$ with bandwidth F_0 can be reconstructed from its sample values $x(n) = x_a(nT_s)$ if the sampling frequency F_s is:

$$F_S \stackrel{\Delta}{=} \frac{1}{T_S} > 2F_0$$

- If $F_S = 2F_0$ we are at the Nyquist rate
- If $F_S < 2F_0$ alias would result in x(n)

Exercise 3 (3/9)

- Frequencies greater than $F_0 > \frac{F_S}{2}$ are seen as frequencies in the base interval (aliasing).
- Ex 3. Given two sinusoidal signals sampled with $F_S = 20$ Hz, one with frequency $F_1 < F_S/2$ (not aliased) and the other with frequency $F_2 = F_1 + F_S$, it could be verified that the two sinusoids samples are perfectly overlapped (aliasing).

Exercise 3 (4/9)

- Pseudocode:
  - fs=20; % sampling frequency
  - t=[0:1/fs:2]; % temporal axis
  - f1=2; % not aliased frequency
  - f2=f1+fs; % aliased frequency

  - plot(t,sin(2*pi*f1*t),'*',t,sin(2*pi*f2*t),'o')

  - t1=[0:1/(fs*10):2];
  - plot(t1,sin(2*pi*f1*t1),t1,sin(2*pi*f2*t1),t, sin(2*pi*f1*t),'*‘)

Exercise 3 (5/9)

- In a strict sense is not possible to analyze analog signal using Matlab. However if we sample $x_a(t)$ in a fine grid $\Delta t << T$, then we can approximate this analysis.

$$x_G(m) = x_a(m\Delta t)$$

- Then: $$X_a(j\Omega) \approx \sum_{m=-\infty}^{\infty} x_G(m) e^{-j\Omega m\Delta t} \Delta t$$

- where Ω is an analog frequency in radians/sec.

Exercise 3 (6/9)

- **EXAMPLE 3.17** Let $x_a(t) = e^{-1000|t|}$. Determine and plot its Fourier transform.

Solution From (3.24)

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t)e^{-j\Omega t}dt = \int_{-\infty}^{0} e^{1000t}e^{-j\Omega t}dt + \int_{0}^{\infty} e^{-1000t}e^{-j\Omega t}dt$$

$$= \frac{0.002}{1+\left(\frac{\Omega}{1000}\right)^2} \qquad (3.32)$$

which is a real-valued function since $x_a(t)$ is a real and even signal. To evaluate $X_a(j\Omega)$ numerically, we have to first approximate $x_a(t)$ by a finite-duration grid sequence $x_G(m)$. Using the approximation $e^{-5} \approx 0$, we note that $x_a(t)$ can be approximated by a finite-duration signal over $-0.005 \le t \le 0.005$ (or equivalently, over $[-5, 5]$ msec). Similarly from (3.32), $X_a(j\Omega) \approx 0$ for $\Omega \ge 2\pi\,(2000)$. Hence choosing

$$\Delta t = 5 \times 10^{-5} \ll \frac{1}{2\,(2000)} = 25 \times 10^{-5}$$

we can obtain $x_G(m)$ and then implement (3.31) in MATLAB.

Exercise 3 (7/9)

Remember that we can compute

$$X_a(j\Omega) \approx \sum_{m=-\infty}^{\infty} x_G(m)\, e^{-j\Omega m\Delta t} \Delta t$$

as $\qquad \overrightarrow{X} = W \overrightarrow{x}$

with $\qquad W = \left[e^{-j\,\overrightarrow{m\Delta t}^{T}\,\overrightarrow{\Omega}} \right]$

Exercise 3 (8/9)

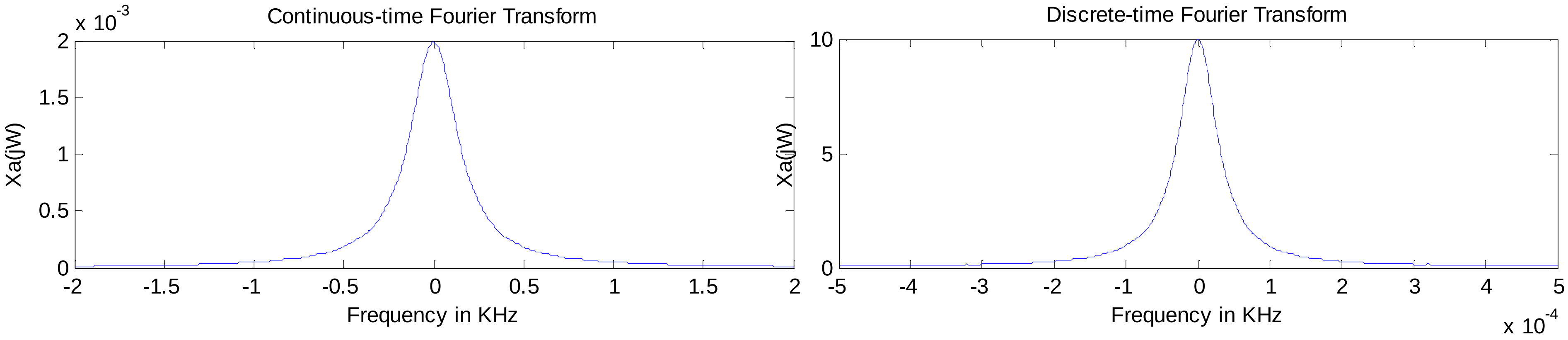
EXAMPLE 3.18 To study the effect of sampling on the frequency-domain quantities, we will sample $x_a(t)$ in Example 3.17 at two different sampling frequencies.

a. Sample $x_a(t)$ at $F_s = 5000$ sam/sec to obtain $x_1(n)$. Determine and plot $X_1(e^{j\omega})$.

b. Sample $x_a(t)$ at $F_s = 1000$ sam/sec to obtain $x_2(n)$. Determine and plot $X_2(e^{j\omega})$.

- N.B. The bandwith of $x_a(t)$ is 2kHz, so if Fs is >4000samp/sec, then aliasing will be almost nonexistent

Exercise 3 (9/9)



- Note hat:
  - the second one amplitude scaled $X\left(e^{j\omega}\right) \propto \frac{1}{T_s} X_a[j\Omega]$
    2*(10^-3)*5000=10
  - the second one is frequency scaled $X\left(e^{j\omega}\right) \rightarrow X_a\left[j\left(\frac{\omega}{T_s}\right)\right]$
    0.5/5000 = 1.0000e-004
  - there's not aliasing

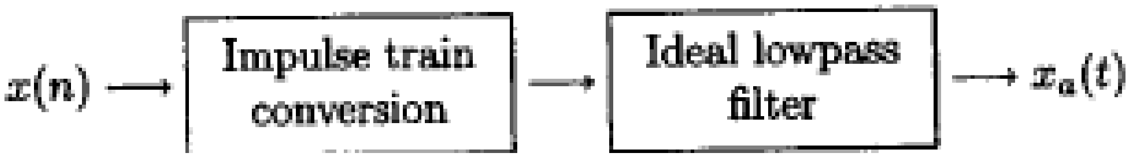
Summary:

- Ex4: Reconstruction
  - Ideal D/A converter
  - Zero-order-hold interpolation
  - First-order-hold interpolation
  - Cubic-spline interpolation

Exercise 4 (12/13)

- If we sample a band-limited signal $x_a(t)$ above its Nyquist rate, then we can reconstruct $x_a(t)$ from its samples $x(n)$.

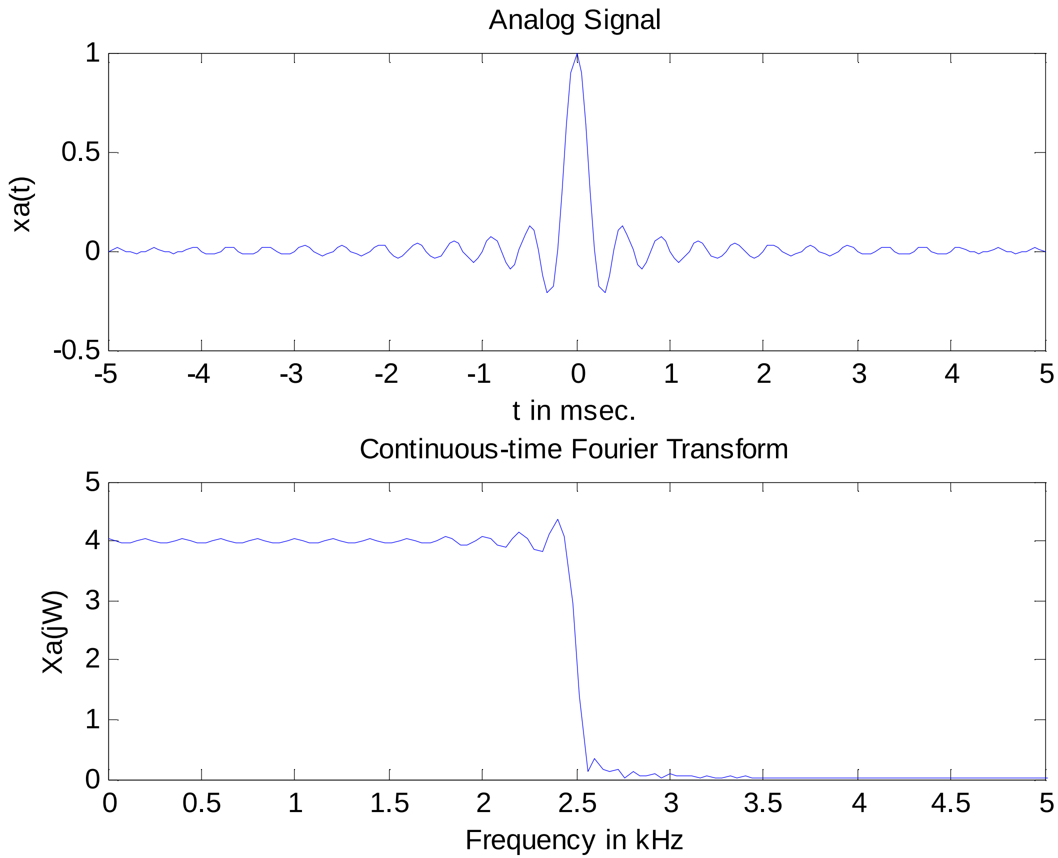
- Reconstruction:



  - The samples are converted into a weighted impulse train
  - The impulse train is filtered trough an ideal analog low pass filter band-limited to $[-F_s/2, F_s/2]$ band.

Exercise 4 (12/13)

- Ideal analog low pass filter with $[-F_s/2, F_s/2]$ band.



$$sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

$$x_a(t) = \sum_{n=-\infty}^{\infty} x(n)\, sinc\left[F_s\left(t - nT_s\right)\right]$$

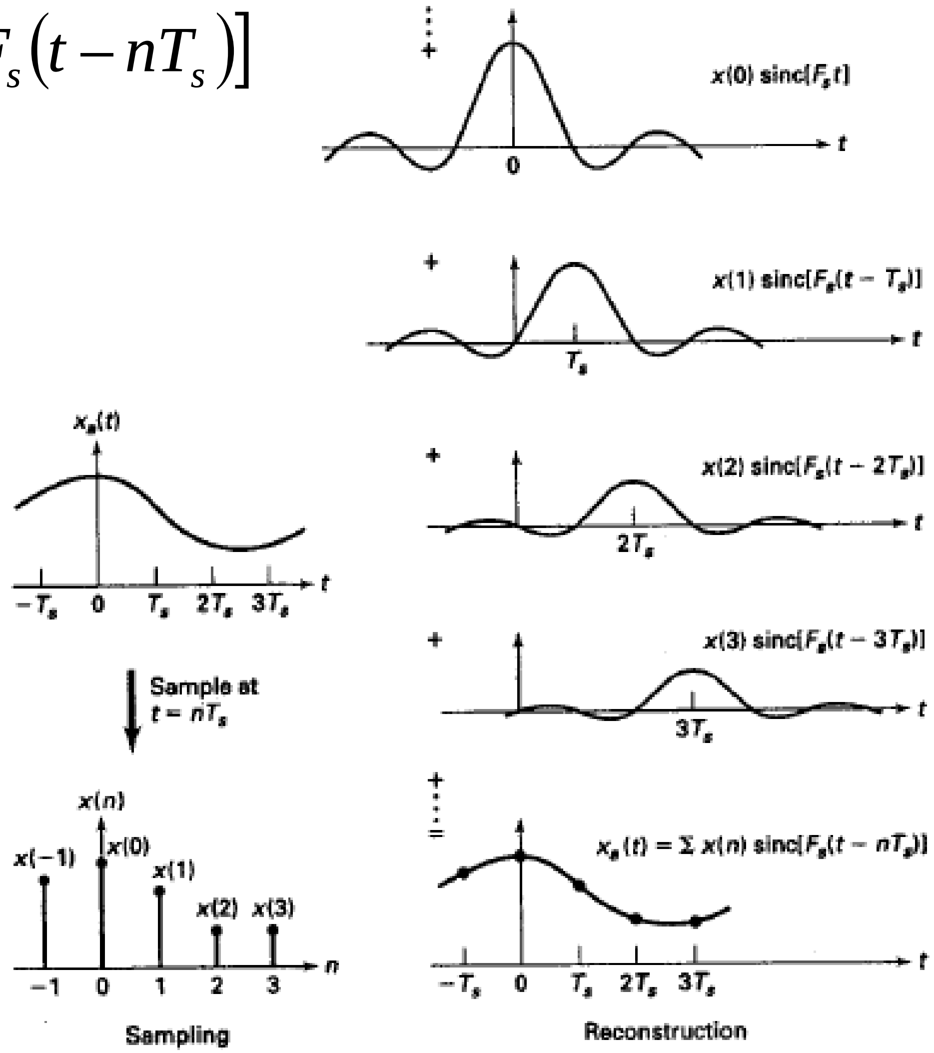


FIGURE 3.14 *Reconstruction of band-limited signal from its samples*

$$x_a(m\Delta t) \approx \sum_{n=n_1}^{n_2} x(n)\, sinc\left[F_s\left(m\Delta t - nT_s\right)\right] \quad t_1 \le m\Delta t \le t_2$$

Exercise 4 (12/13)

□ **EXAMPLE 3.19** From the samples $x_1(n)$ in Example 3.18a, reconstruct $x_a(t)$ and comment on the results.

Solution Note that $x_1(n)$ was obtained by sampling $x_a(t)$ at $T_s = 1/F_s = 0.0002$ sec. We will use the grid spacing of 0.00005 sec over $-0.005 \le t \le 0.005$, which gives $x(n)$ over $-25 \le n \le 25$.

Remember that $x_a(t) = e^{-1000|t|}$

grid sequence $x_G(m)$. Using the approximation $e^{-5} \approx 0$, we note that $x_a(t)$ can be approximated by a finite-duration signal over $-0.005 \le t \le 0.005$ (or equivalently, over $[-5, 5]$ msec). Similarly from (3.32), $X_a(j\Omega) \approx 0$ for $\Omega \ge 2\pi(2000)$. Hence choosing

$$\Delta t = 5 \times 10^{-5} \ll \frac{1}{2(2000)} = 25 \times 10^{-5}$$

and that in the example3.18 we have sampled at Fs=5000sam/sec

Exercise 4 (12/13)

$$x_a(t) = e^{-1000|t|}$$

- **EXAMPLE 3.20** From the samples $x_2(n)$ in Example 3.18b reconstruct $x_a(t)$ and comment on the results.

In the example3.18b we have sampled at Fs=1000sam/sec

Exercise 4 (12/13)

- Zero-order-hold (ZOH) interpolation:

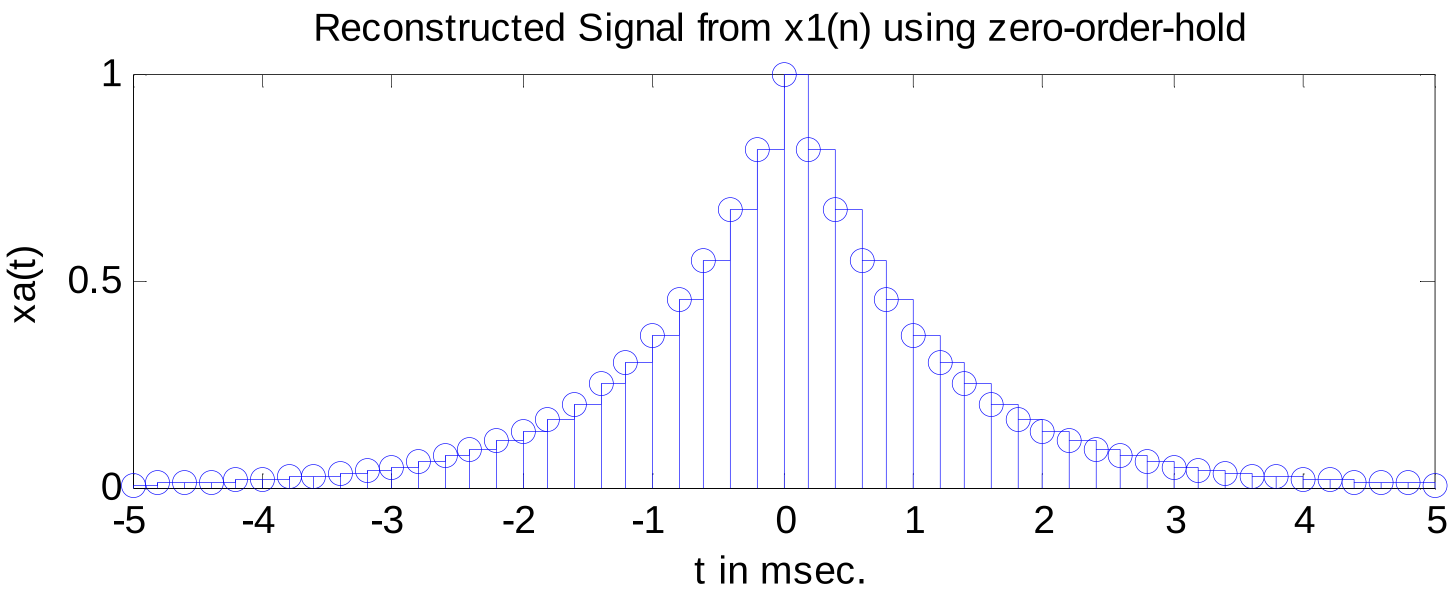
$$\hat{x}_a(t) = x(n) \quad nT_s \le n < (n+1)T_s$$

  - A given sample value is held for the sample interval until the next sample is received

$$h_0(t) = \begin{cases} 1 & 0 \le t \le T_s \\ 0 & otherwise \end{cases}$$

Exercise 4 (12/13)

- The resulting signal is piece-wise constant



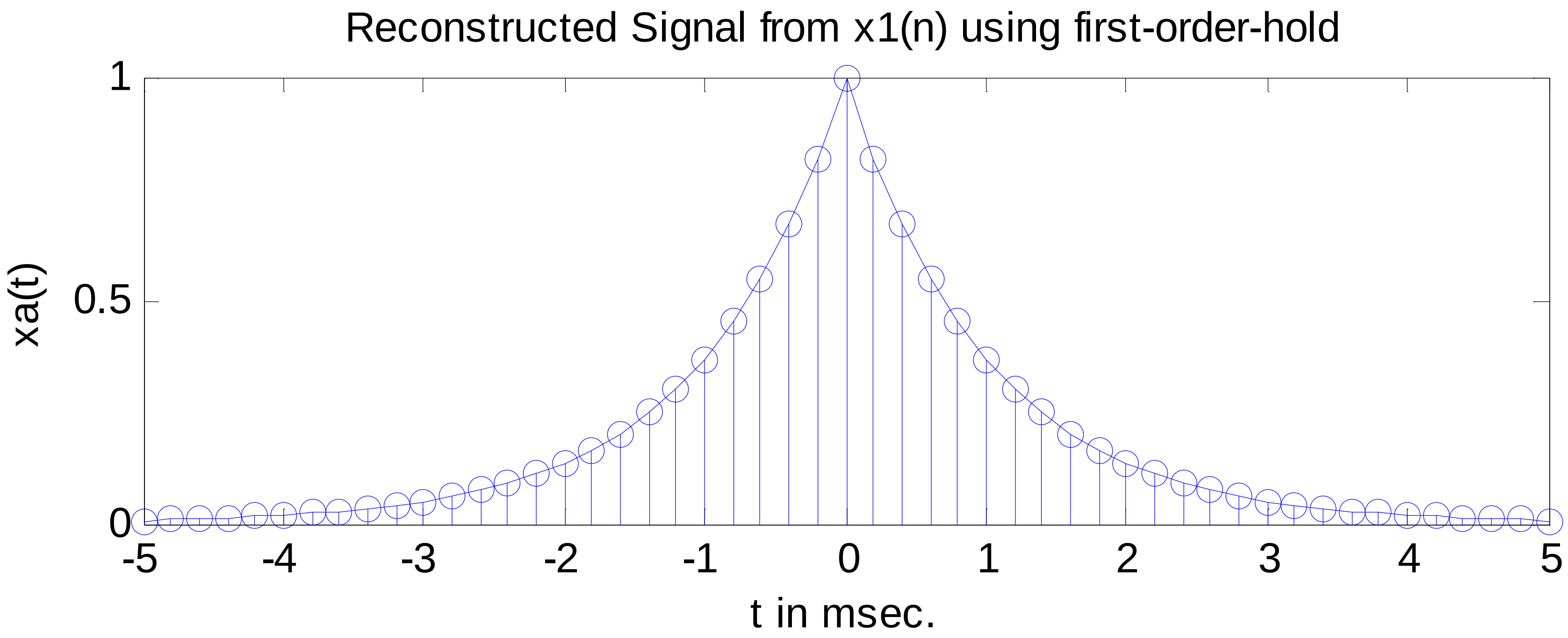
Exercise 4 (12/13)

- First-order-hold (FOH) interpolation:
  - Adjacent samples are joined by straight lines

$$h_1(t) = \begin{cases} 1 + \dfrac{t}{T_s} & 0 \le t \le T_s \\ 1 - \dfrac{t}{T_s} & T_s \le t \le 2T_s \\ 0 & otherwise \end{cases}$$

Exercise 4 (12/13)

- The resulting signal is linear-wise constant



Exercise 4 (12/13)

- Cubic spline interpolation:

$$\hat{x}_a(t) = \alpha_0(n) + \alpha_1(n)(t - nT_s) + \alpha_2(n)(t - nT_s)^2 + \\ + \alpha_3(n)(t - nT_s)^3 \quad nT_s \le n < (n+1)T_s$$

  - Where αi(n) are the polynomial coefficients which are determinated by using least-squares analysis on the sample values.
  - Matlab provides the function “xa=spline(nTs,x,t), where x and nTs are arrays containing samples x(n) at nTs instances, and t array contains a finer grid at which xa(t) values are desired.

Exercise 4 (12/13)

EXAMPLE 3.22 From the samples $x_1(n)$ and $x_2(n)$ in Example 3.18, reconstruct $x_a(t)$ using the spline function. Comment on the results.