Digital Signal Processing

Prof. Nizamettin AYDIN, PhD

naydin@yildiz.edu.tr
nizamettinaydin@gmail.com
nizamettinaydin@aydin.edu.tr
http://www3.yildiz.edu.tr/~naydin

1

1

Sampling & Aliasing

2

2

LECTURE OBJECTIVES

- SAMPLING can cause ALIASING
  - Sampling Theorem
  - Sampling Rate > 2(Highest Frequency)
- Spectrum for digital signals, $x[n]$
  - Normalized Frequency

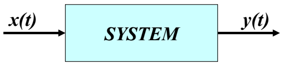
$$\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s} + 2\pi\ell$$

ALIASING

3

3

SYSTEMS Process Signals



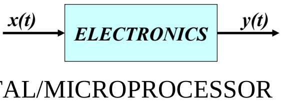
- PROCESSING GOALS:
  - Change $x(t)$ into $y(t)$
    - For example, more BASS
  - Improve $x(t)$, e.g., image deblurring
  - Extract Information from $x(t)$

4

4

System IMPLEMENTATION

- ANALOG/ELECTRONIC:
    - Circuits: resistors, capacitors, op-amps



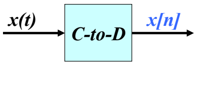
- DIGITAL/MICROPROCESSOR
    - Convert $x(t)$ to numbers stored in memory

5

5

SAMPLING $x(t)$

- SAMPLING PROCESS
    - Convert $x(t)$ to numbers $x[n]$
    - "n" is an integer; $x[n]$ is a sequence of values
    - Think of "n" as the storage address in memory
- UNIFORM SAMPLING at $t = nT_s$
    - IDEAL: $x[n] = x(nT_s)$

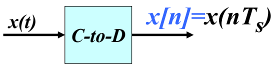


6

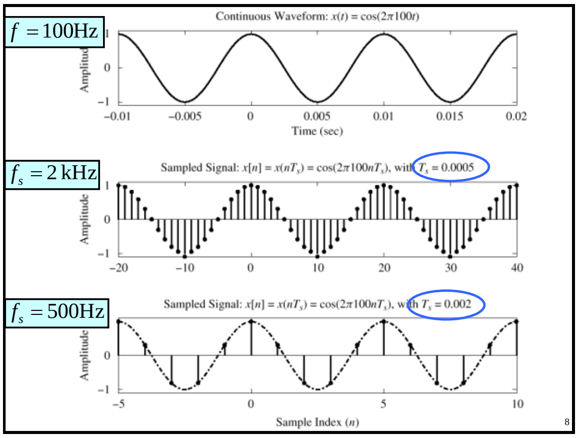
6

SAMPLING RATE, f_s

- SAMPLING RATE (f_s)
  - $f_s = 1/T_s$
    - NUMBER of SAMPLES PER SECOND
  - T_s = 125 microsec → f_s = 8000 samples/sec
    - UNITS ARE HERTZ: 8000 Hz
- UNIFORM SAMPLING at $t = nT_s = n/f_s$
  - IDEAL: $x[n] = x(nT_s) = x(n/f_s)$



7



8

SAMPLING THEOREM

- HOW OFTEN ?
  - DEPENDS on FREQUENCY of SINUSOID
  - ANSWERED by SHANNON/NYQUIST Theorem
  - ALSO DEPENDS on "**RECONSTRUCTION**"

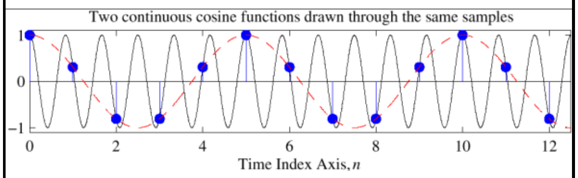
Shannon Sampling Theorem

A continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n] = x(nT_s)$, if the samples are taken at a rate $f_s = 1/T_s$ that is greater than $2f_{max}$.

9

Reconstruction? Which One?

Given the samples, draw a sinusoid through the values



$$x[n] = \cos(0.4\pi n)$$

When n is an integer $\cos(0.4\pi n) = \cos(2.4\pi n)$

10

STORING DIGITAL SOUND

- **x[n]** is a SAMPLED SINUSOID
  - A list of numbers stored in memory
- EXAMPLE: audio CD
- CD rate is 44,100 samples per second
  - 16-bit samples
  - Stereo uses 2 channels
- Number of bytes for 1 minute is
  - 2 × (16/8) × 60 × 44100 = 10.584 Mbytes

11

DISCRETE-TIME SINUSOID

- Change x(t) into x[n] **DERIVATION**

$$x(t) = A\cos(\omega t + \varphi)$$
$$x[n] = x(nT_s) = A\cos(\omega nT_s + \varphi)$$
$$x[n] = A\cos((\omega T_s)n + \varphi)$$
$$x[n] = A\cos(\hat{\omega} n + \varphi)$$
$$\hat{\omega} = \omega T_s = \frac{\omega}{f_s}$$

DEFINE DIGITAL FREQUENCY

12

DIGITAL FREQUENCY $\hat{\omega}$

- $\hat{\omega}$ VARIES from **0** to **2π**, as *f* varies from 0 to the sampling frequency
- UNITS are radians, **not** rad/sec
  - DIGITAL FREQUENCY is NORMALIZED

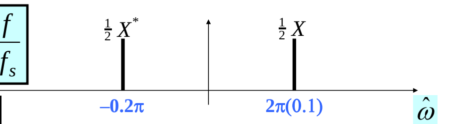
$$\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s}$$

13

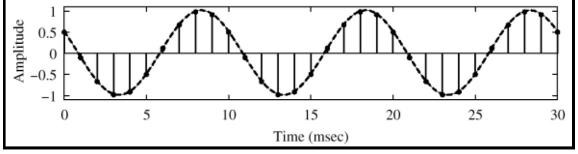
SPECTRUM (DIGITAL)

$\hat{\omega} = 2\pi \frac{f}{f_s}$

$f_s = 1\,\text{kHz}$



$x[n] = A\cos(2\pi(100)(n/1000) + \varphi)$

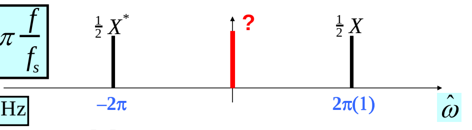


14

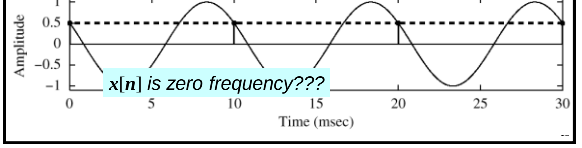
SPECTRUM (DIGITAL) ???

$\hat{\omega} = 2\pi \frac{f}{f_s}$

$f_s = 100\,\text{Hz}$



$x[n] = A\cos(2\pi(100)(n/100) + \varphi)$



15

The REST of the STORY

- Spectrum of x[n] has more than one line for each complex exponential
  - Called **ALIASING**
  - **MANY SPECTRAL LINES**
- SPECTRUM is PERIODIC with period = **2π**
  - Because

$$A\cos(\hat{\omega}n + \varphi) = A\cos((\hat{\omega} + 2\pi)n + \varphi)$$

16

ALIASING DERIVATION

- Other Frequencies give the same $\hat{\omega}$

$x_1(t) = \cos(400\pi t)$ sampled at $f_s = 1000\,\text{Hz}$

$x_1[n] = \cos(400\pi \tfrac{n}{1000}) = \cos(0.4\pi n)$

$x_2(t) = \cos(2400\pi t)$ sampled at $f_s = 1000\,\text{Hz}$

$x_2[n] = \cos(2400\pi \tfrac{n}{1000}) = \cos(2.4\pi n)$

$x_2[n] = \cos(2.4\pi n) = \cos(0.4\pi n + 2\pi n) = \cos(0.4\pi n)$

$\Rightarrow x_2[n] = x_1[n]$ $\quad 2400\pi - 400\pi = 2\pi(1000)$

17

ALIASING DERIVATION–2

- Other Frequencies give the same $\hat{\omega}$

If $x(t) = A\cos(2\pi(f + \ell f_s)t + \varphi)$ $\quad t \leftarrow \frac{n}{f_s}$

and we want : $x[n] = A\cos(\hat{\omega}n + \varphi)$

then : $\hat{\omega} = \dfrac{2\pi(f + \ell f_s)}{f_s} = \dfrac{2\pi f}{f_s} + \dfrac{2\pi \ell f_s}{f_s}$

$$\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s} + 2\pi\ell$$

18

ALIASING CONCLUSIONS

- ADDING f_s or $2f_s$ or $-f_s$ to the FREQ of $x(t)$ gives exactly the same $x[n]$
  - The samples, $x[n] = x(n/f_s)$ are EXACTLY THE SAME VALUES
- GIVEN $x[n]$, WE CAN'T DISTINGUISH f_o FROM $(f_o + f_s)$ or $(f_o + 2f_s)$

19

19

NORMALIZED FREQUENCY

- DIGITAL FREQUENCY

Normalized Radian Frequency

$$\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s} + 2\pi\ell$$

Normalized Cyclic Frequency

$$\hat{f} = \hat{\omega}/(2\pi) = f T_s = f/f_s$$

20

20

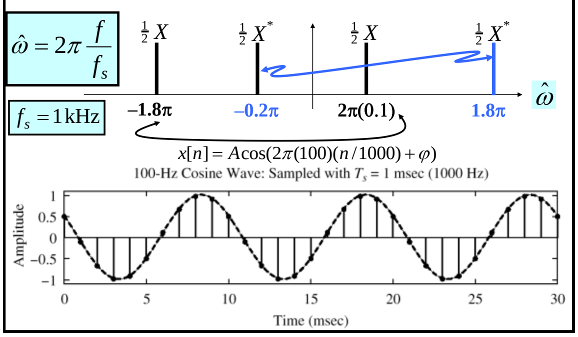
SPECTRUM for $x[n]$

- PLOT versus NORMALIZED FREQUENCY
- INCLUDE **ALL** SPECTRUM LINES
  - ALIASES
    - ADD MULTIPLES of 2π
    - SUBTRACT MULTIPLES of 2π
  - FOLDED ALIASES
    - ALIASES of NEGATIVE FREQS

21

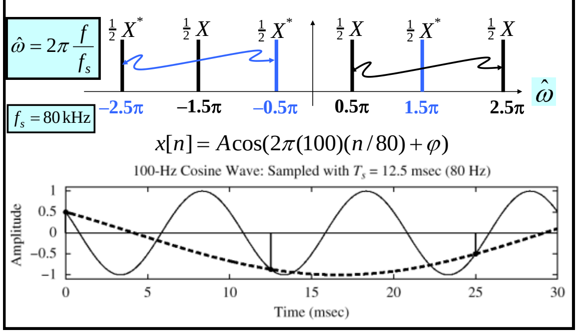
21

SPECTRUM (MORE LINES)



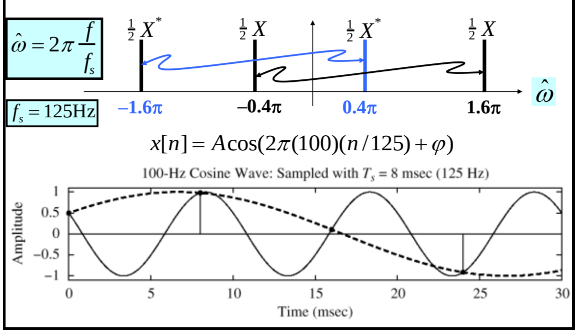
22

SPECTRUM (ALIASING CASE)



23

SPECTRUM (FOLDING CASE)



24

D-to-A Conversion

33

33

SIGNAL TYPES

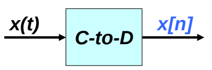
- A-to-D
  - Convert $x(t)$ to numbers stored in memory
- D-to-A
  - Convert $y[n]$ back to a "continuous-time" signal, $y(t)$
  - $y[n]$ is called a "discrete-time" signal

34

34

SAMPLING $x(t)$

- UNIFORM SAMPLING at $t = nT_s$
  - IDEAL: $x[n] = x(nT_s)$



Shannon Sampling Theorem

A continuous-time signal $x(t)$ with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n] = x(nT_s)$, if the samples are taken at a rate $f_s = 1/T_s$ that is greater than $2f_{max}$.

35

35

NYQUIST RATE

- "Nyquist Rate" Sampling
  - f_s > **TWICE** the HIGHEST Frequency in $x(t)$
  - "Sampling above the Nyquist rate"
- **BANDLIMITED SIGNALS**
  - DEF: x(t) has a HIGHEST FREQUENCY COMPONENT in its SPECTRUM
  - NON-BANDLIMITED EXAMPLE
    - TRIANGLE WAVE is NOT BANDLIMITED

36

36

SPECTRUM for $x[n]$

- INCLUDE ALL SPECTRUM LINES
  - ALIASES
    - ADD INTEGER MULTIPLES of 2π and -2π
  - FOLDED ALIASES
    - ALIASES of NEGATIVE FREQS
- PLOT versus NORMALIZED FREQUENCY
  - i.e., DIVIDE f_o by f_s

$$\hat{\omega} = 2\pi \frac{f}{f_s} + 2\pi\ell$$

37

37

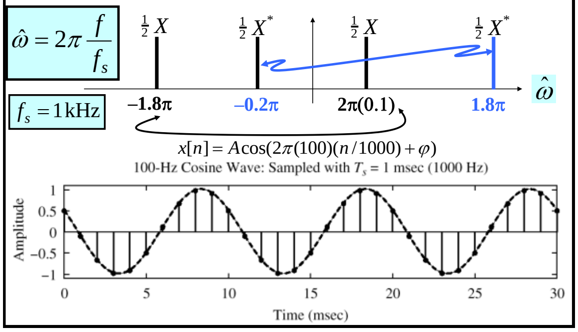
EXAMPLE: SPECTRUM

- $x[n] = A\cos(0.2\pi n + \phi)$
- FREQS @ 0.2π and -0.2π
- ALIASES:
  - $\{2.2\pi, 4.2\pi, 6.2\pi, \ldots\}$ & $\{-1.8\pi, -3.8\pi, \ldots\}$
  - EX: $x[n] = A\cos(4.2\pi n + \phi)$
- ALIASES of NEGATIVE FREQ:
  - $\{1.8\pi, 3.8\pi, 5.8\pi, \ldots\}$ & $\{-2.2\pi, -4.2\pi \ldots\}$

38

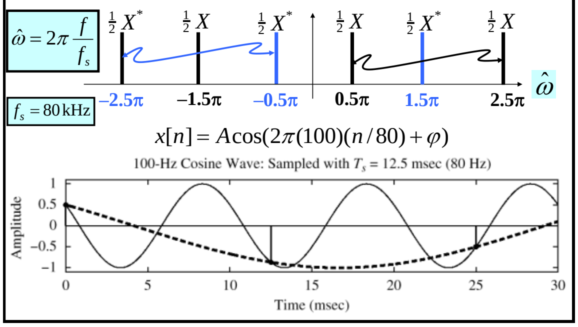
38

SPECTRUM (MORE LINES)



39

SPECTRUM (ALIASING CASE)



40

FOLDING (a type of ALIASING)

- EXAMPLE: 3 different $x(t)$; same $x[n]$

$$\hat{\omega} = 2\pi \frac{100}{1000} = 2\pi(0.1)$$

$f_s = 1000$

$\cos(2\pi(100)t) \rightarrow \cos[2\pi(0.1)n]$

$\cos(2\pi(1100)t) \rightarrow \cos[2\pi(1.1)n] = \cos[2\pi(0.1)n]$

$\cos(2\pi(900)t) \rightarrow \cos[2\pi(0.9)n]$

$= \cos[2\pi(0.9)n - 2\pi n] = \cos[2\pi(-0.1)n] = \cos[2\pi(0.1)n]$

- 900 Hz "folds" to 100 Hz when f_s=1kHz

41

41

DIGITAL FREQ $\hat{\omega}$ AGAIN

Normalized Radian Frequency

$$\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s} + 2\pi\ell$$

ALIASING

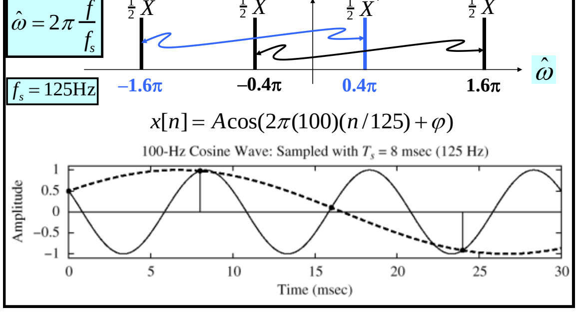
$$\hat{\omega} = \omega T_s = -\frac{2\pi f}{f_s} + 2\pi\ell$$

FOLDED ALIAS

42

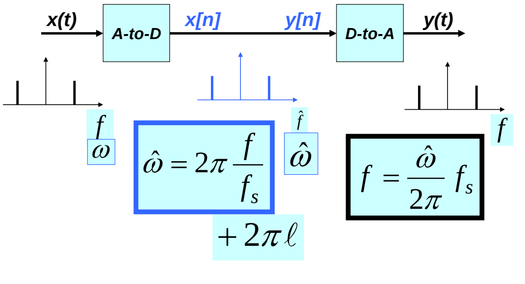
42

SPECTRUM (FOLDING CASE)



43

FREQUENCY DOMAINS



44

44

D-to-A Reconstruction

- Create continuous $y(t)$ from $y[n]$
  - **IDEAL**
    - If you have formula for $y[n]$
  - Replace n in $y[n]$ with $f_s t$
  - $y[n] = A\cos(0.2\pi n+\phi)$ with $f_s = 8000$ Hz
  - $y(t) = A\cos(2\pi(800)t+\phi)$

46

46

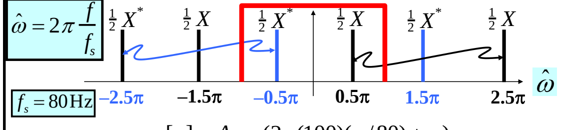
D-to-A is AMBIGUOUS !

- ALIASING
  - Given $y[n]$, which $y(t)$ do we pick ? ? ?
  - INFINITE NUMBER of $y(t)$
    - PASSING THRU THE SAMPLES, $y[n]$
  - D-to-A RECONSTRUCTION MUST CHOOSE ONE OUTPUT
- RECONSTRUCT THE SMOOTHEST ONE
  - THE LOWEST FREQ, if $y[n]$ = sinusoid

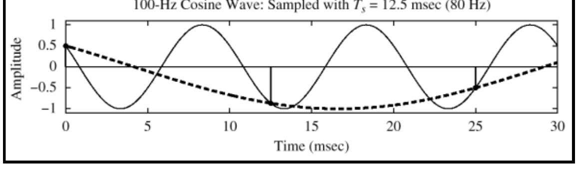
47

47

SPECTRUM (ALIASING CASE)



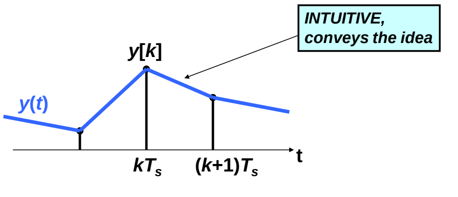
$$x[n] = A\cos(2\pi(100)(n/80)+\varphi)$$



48

Reconstruction (D-to-A)

- CONVERT STREAM of NUMBERS to $x(t)$
- "CONNECT THE DOTS"
- INTERPOLATION

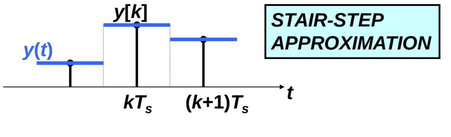


49

49

SAMPLE & HOLD DEVICE

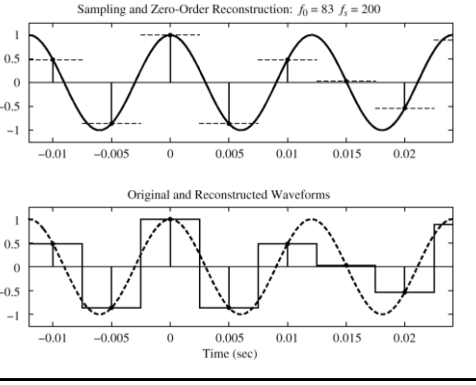
- CONVERT $y[n]$ to $y(t)$
  - $y[k]$ should be the value of $y(t)$ at $t = kT_s$
  - Make $y(t)$ equal to $y[k]$ for
    - $kT_s - 0.5T_s < t < kT_s + 0.5T_s$



50

50

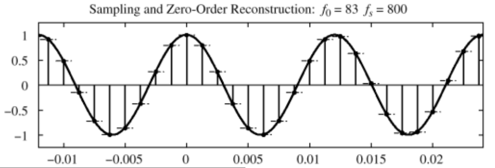
SQUARE PULSE CASE



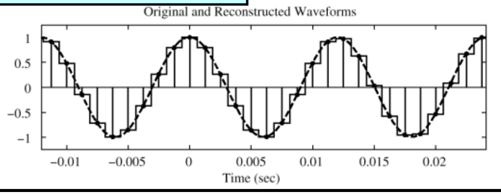
51

51

OVER-SAMPLING CASE



EASIER TO RECONSTRUCT



52

52

MATH MODEL for D-to-A

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]p(t - nT_s)$$

SQUARE PULSE:

$$p(t) = \begin{cases} 1 & -\frac{1}{2}T_s < t \le \frac{1}{2}T_s \\ 0 & \text{otherwise} \end{cases}$$

53

53

EXPAND the SUMMATION

$$\sum_{n=-\infty}^{\infty} y[n]p(t - nT_s) =$$

$$\ldots + y[0]p(t) + y[1]p(t - T_s) + y[2]p(t - 2T_s) + \ldots$$

- SUM of SHIFTED PULSES $p(t\text{-}nT_s)$
  - "WEIGHTED" by $y[n]$
  - CENTERED at $t=nT_s$
  - SPACED by T_s
    - RESTORES "REAL TIME"

54

54

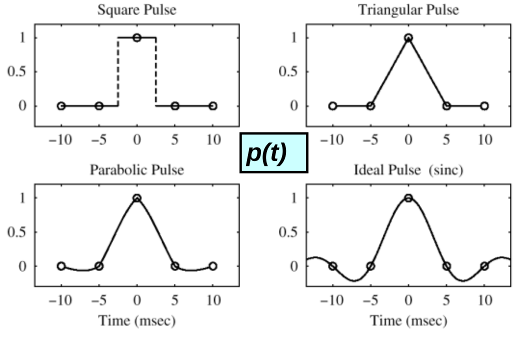


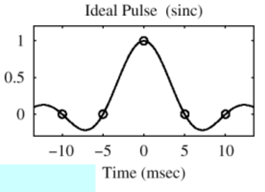
Figure 4.17 Four different pulses for D-to-C conversion. The sampling period is $T_s = 0.005$, i.e., $f_s = 200$ Hz. Note that the duration of each pulse is approximately one or two times T_s.

55

55

OPTIMAL PULSE ?

CALLED "BANDLIMITED INTERPOLATION"



$$p(t) = \frac{\sin \frac{\pi t}{T_s}}{\frac{\pi t}{T_s}} \quad \text{for } -\infty < t < \infty$$

$$p(t) = 0 \quad \text{for } \ t = \pm T_s, \pm 2T_s, \ldots$$

56

56