DSP First, 2/e

MODIFIED TLH

Lecture 9

Sampling & Aliasing

SAMPLING AND ALIASING REFERENCES CH 4 2/12/2019

Let's Go to the Movies

Interesting illusion: the helicopter's blades are somehow in sync with the camera's shutter making it seem as though they are not moving. 0:47 https://www.youtube.com/watch?v=qgvuQGY946g

Helicopter blades and other fast spinning objects often produce strange effects on camera. Lauren explains why. 3:51

https://www.youtube.com/watch?v=AYQAKwCxScc

Audio Sampling Rate Demo 0:35

https://www.youtube.com/watch?v=hRhVb6iRArg&feature=youtu.be

Video Aliasing

Why car wheels rotate backwards in movies 4:25

https://www.youtube.com/watch?v=SFbINinFsxk&feature=youtu.be

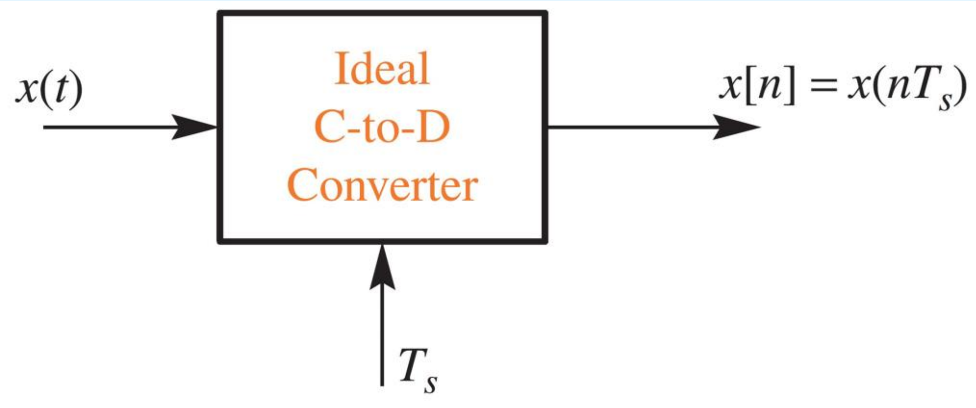
READING ASSIGNMENTS

- This Lecture:
  - Chap 4, Sections 4-1 and 4-2
- Other Reading:
  - TLH WEBSITE
  - Next Lecture: Chap. 4, Sects. 4-3 and 4-4

Sampling

Figure 4-1: Block diagram representation of the ideal continuous-to-discrete (C-to-D) converter. The parameter T_s specifies uniform sampling of the input signal every T_s seconds.

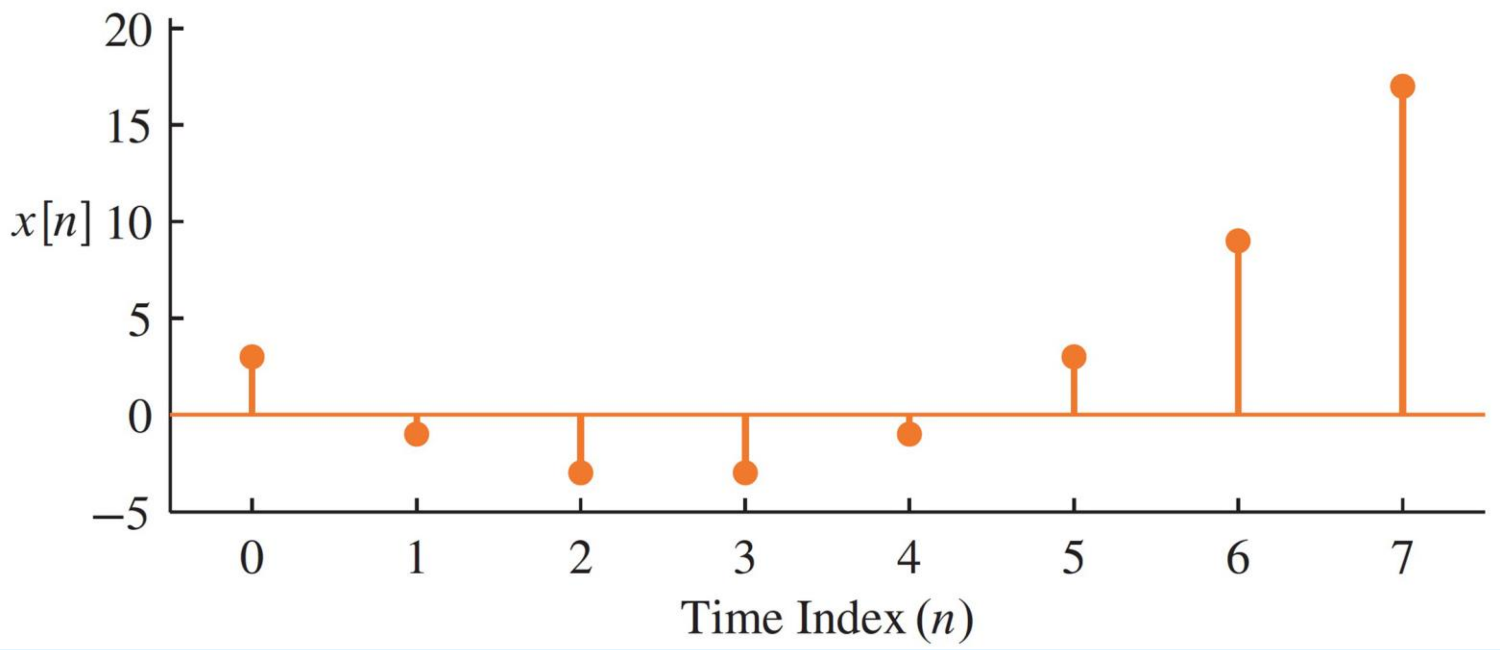
n = 1,2,3,4… nTs in seconds



Sometimes ADC, A2D, Analog-to-Digital

Sampling Sinusoidal Signals (1 of 2)

Figure 4-2: Plotting format for discrete-time signals, called a stem plot. In MATLAB, the function stem produces this plot. Some students also refer to the stem plot as a “lollypop” plot.



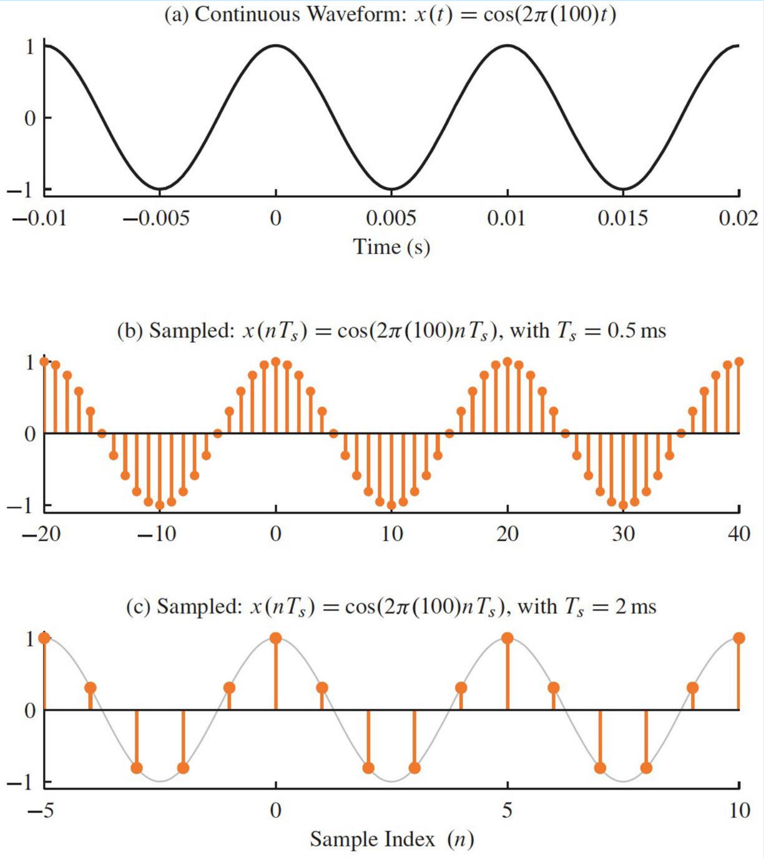
```
>> help stem
 stem   Discrete sequence or "stem" plot.

 stem(Y) plots the data sequence Y as stems from the x axis
terminated with circles for the data value. If Y is a matrix
then  each column is plotted as a separate series.

  stem(X,Y) plots the data sequence Y at the values specified
in X.
```

Sampling Sinusoidal Signals (2 of 2)

Figure 4-3: A continuous-time 100 Hz sinusoid (a) and two discrete-time sinusoids formed by sampling at $f_s = 2000\,\text{samples/s}$ (b) and at $f_s = 500\,\text{samples/s}$ (c).



LECTURE OBJECTIVES

- SAMPLING can cause ALIASING
  - **<u>Sampling Theorem</u>**
  - 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

SYSTEMS Process Signals

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

System IMPLEMENTATION

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

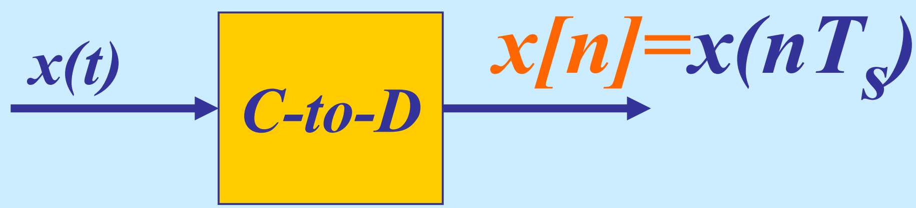
- DIGITAL/MICROPROCESSOR
  - Convert x(t) to numbers stored in memory

SAMPLING x(t)

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

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 of f_s ARE HERTZ: 8000 Hz
- UNIFORM SAMPLING at $t = nT_s = n/f_s$
  - IDEAL: $x[n] = x(nT_s) = x(n/f_s)$



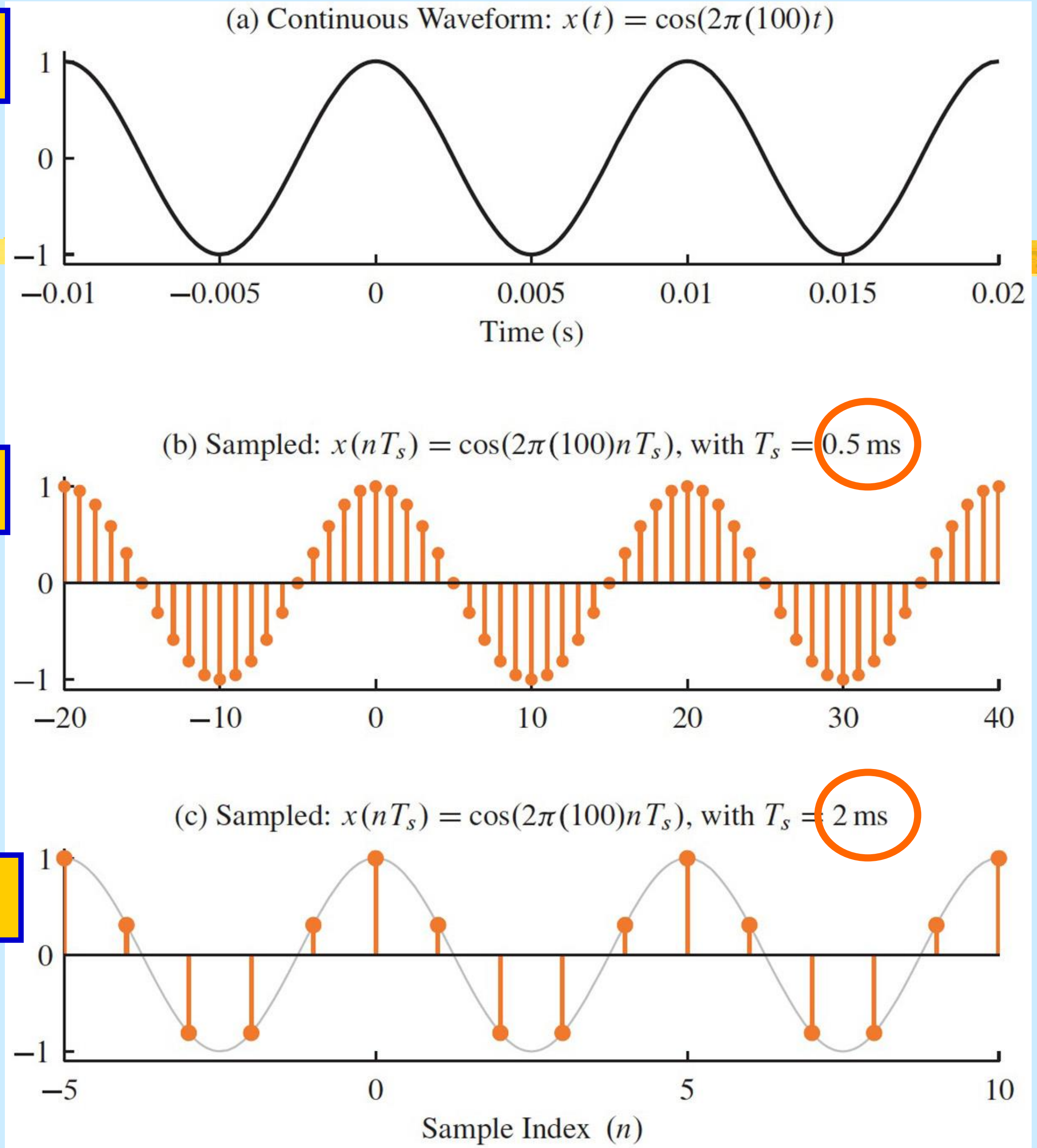
STORING DIGITAL SOUND

- $x[n]$ is a SAMPLED SISIGNAL
  - 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 X (16/8) X 60 X 44100 = 10.584 Mbytes

$f = 100\text{Hz}$

$f_s = 2\ \text{kHz}$

$f_s = 500\text{Hz}$



Which one provides the most accurate representation of x(t)?

SAMPLING THEOREM

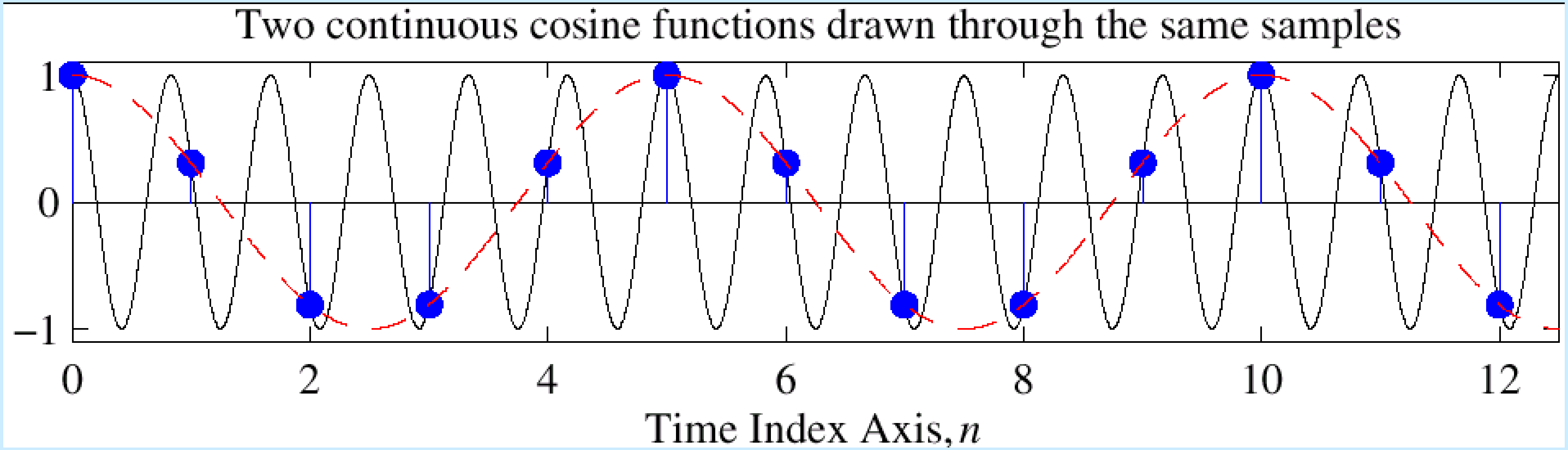
- HOW OFTEN DO WE NEED TO SAMPLE?
  - DEPENDS on FREQUENCY of SINUSOID
  - ANSWERED by SHANNON/NYQUIST Theorem
  - ALSO DEPENDS on "**RECONSTRUCTION**"

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 $2f_{\max}$.

Reconstruction? Which One?

Given the samples, draw a sinusoid through the values



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

When n is an integer

$$\cos(0.4\pi n) = \cos(2.4\pi n)$$

Occam's razor -> pick lowest frequency sinusoid

Be careful...

See the References on the Website

https://www.youtube.com/watch?v=qgvuQGY946g

DISCRETE-TIME SINUSOID

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

$$x(t) = A\cos(\omega t + \varphi)$$

$$x[n] = x(nT_s) = A\cos(\omega n T_s + \varphi)$$

$$x[n] = A\cos((\omega T_s)n + \varphi)$$

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

$$\hat{\omega} = \omega T_s = \frac{\omega}{f_s}$$

DEFINE DIGITAL FREQUENCY

DIGITAL FREQUENCY $\hat{\omega}$

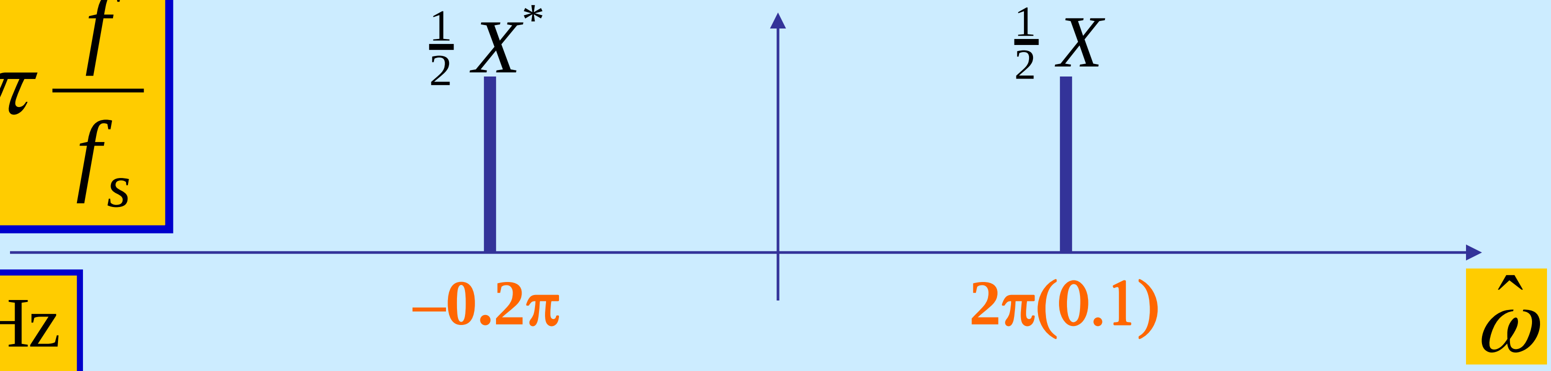
$$\boxed{\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s}}$$

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

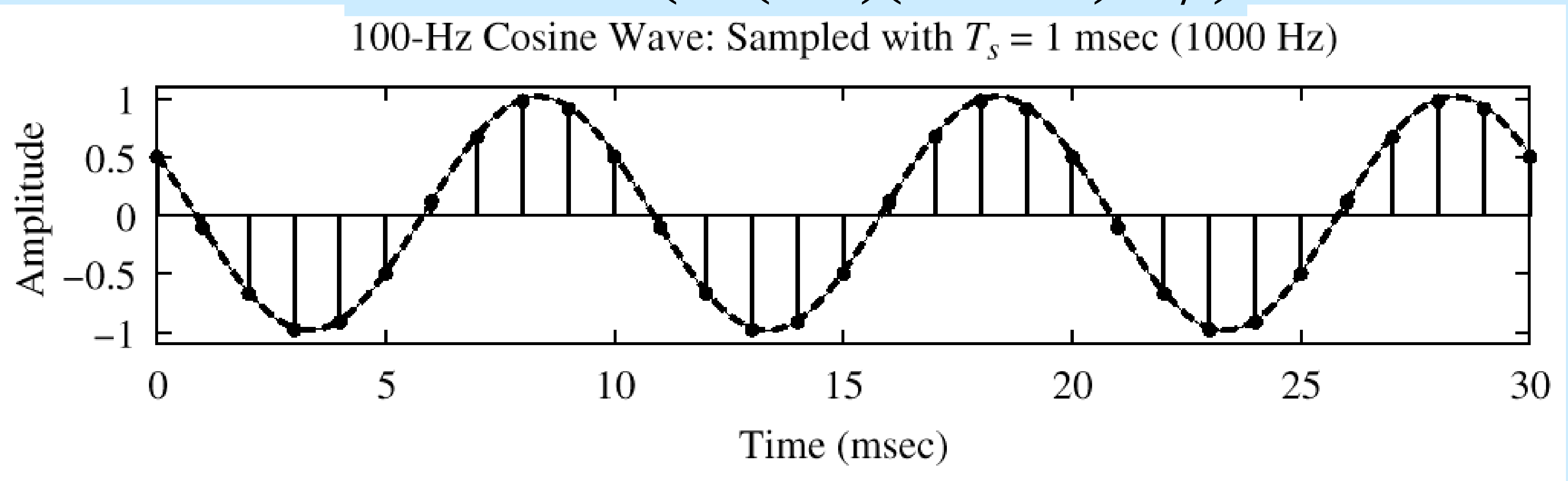
SPECTRUM (DIGITAL)

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

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



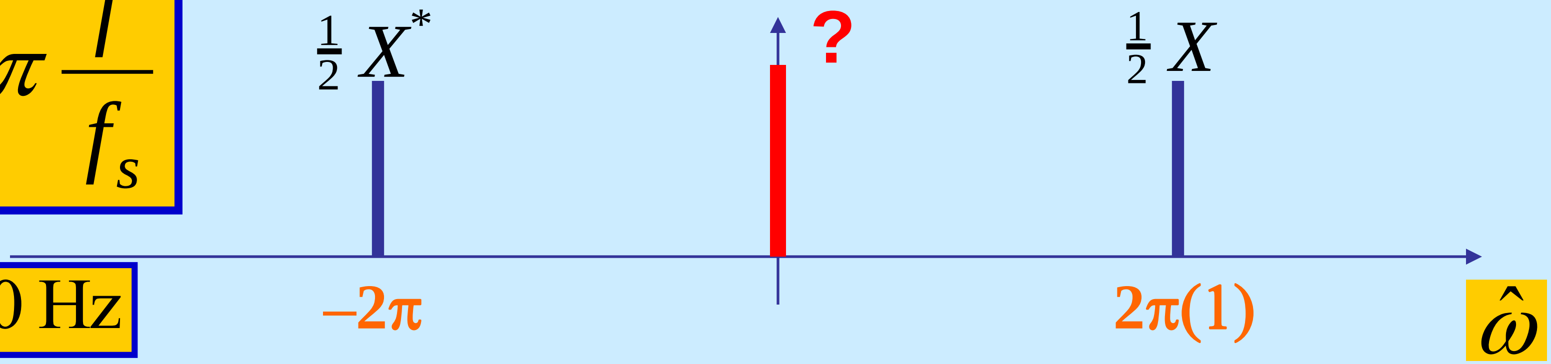
$$x[n] = A\cos(2\pi(100)(n/1000) + \varphi)$$



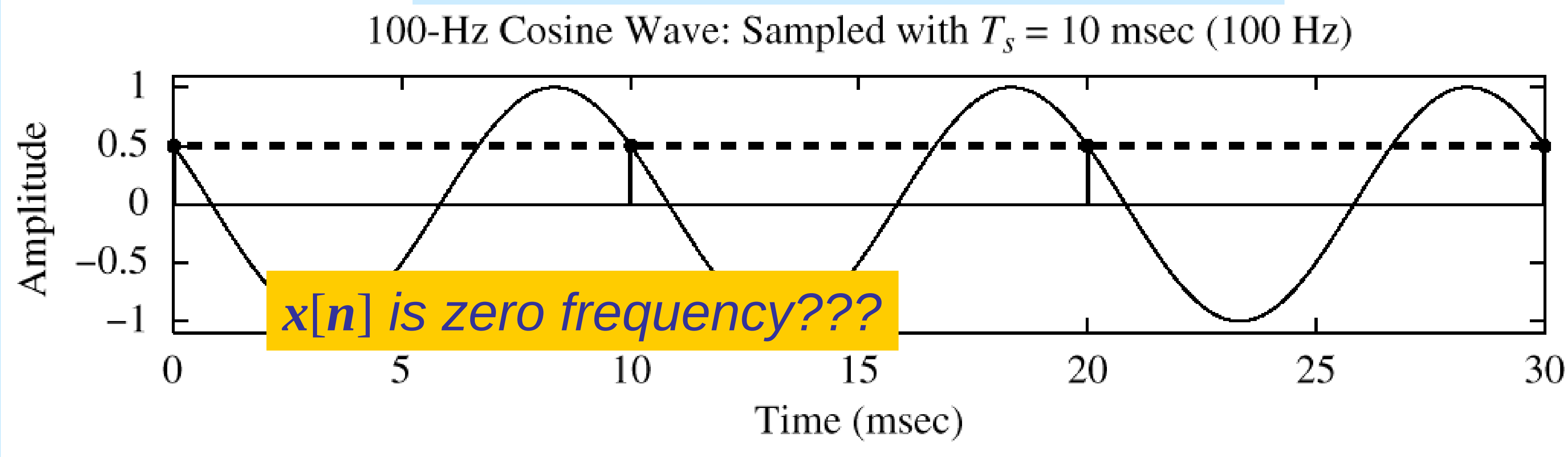
SPECTRUM (DIGITAL) ???

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

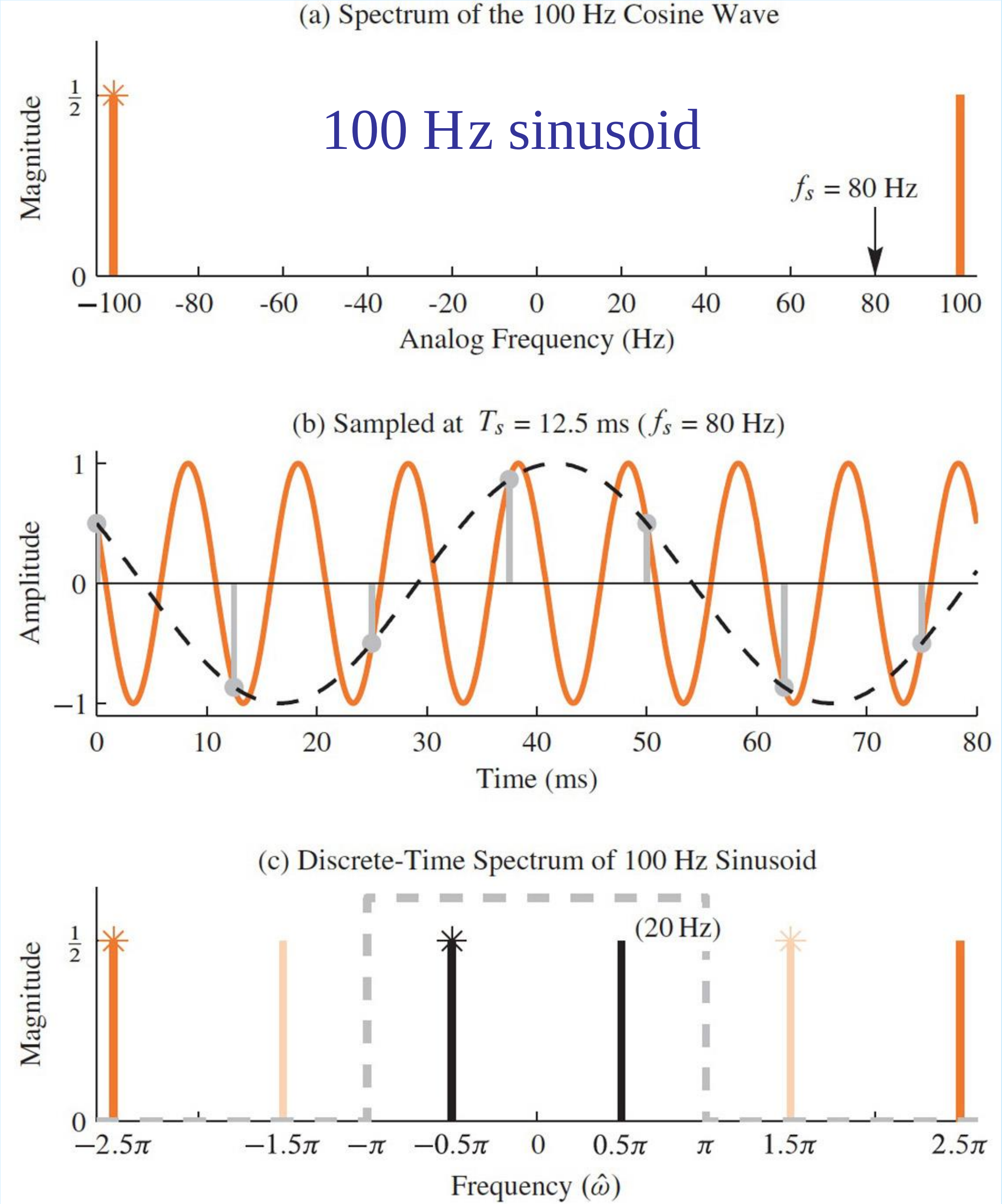
$f_s = 100$ Hz



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



x[n] is zero frequency???



$f_s = 80$ samples/s.

$\hat{\omega} = \pm 2 \cdot 5\pi$ rad,

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

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

ALIASING DERIVATION

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

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

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

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

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

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

$$\Rightarrow x_2[n] = x_1[n]$$

$$2400\pi - 400\pi = 2\pi(1000)$$

ALIASING DERIVATION

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

$$t \leftarrow \frac{n}{f_s}$$

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

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

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

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

ALIASING CONCLUSIONS

- Adding an **INTEGER multiple** of f_s or $–f_s$ to the frequency of a continuous sinusoid $x_c(t)$ gives **exactly the same values** for the sampled signal $x[n] = x_c(n/f_s)$
- **GIVEN x[n], we CAN'T KNOW whether it came from a sinusoid at f_o or $(f_o + f_s)$ or $(f_o + 2f_s)$ …**
- **This is called ALIASING**

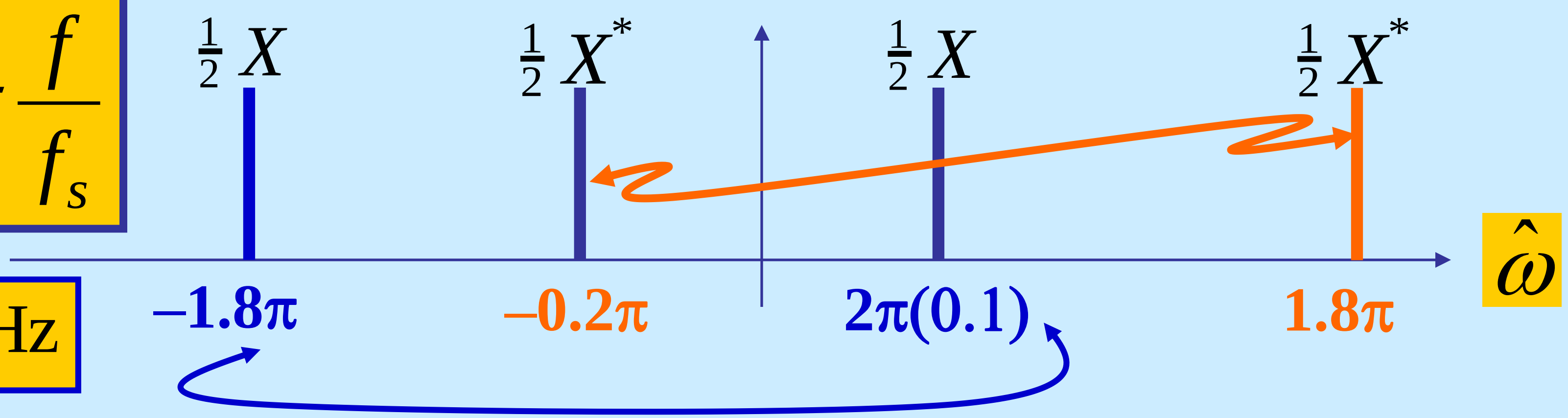
SPECTRUM for x[n]

- PLOT versus NORMALIZED FREQUENCY
- INCLUDE **ALL** SPECTRUM LINES
  - ALIASES
    - ADD MULTIPLES of 2π
    - SUBTRACT MULTIPLES of 2π
  - FOLDED ALIASES
    - (to be discussed later)
    - ALIASES of NEGATIVE FREQS

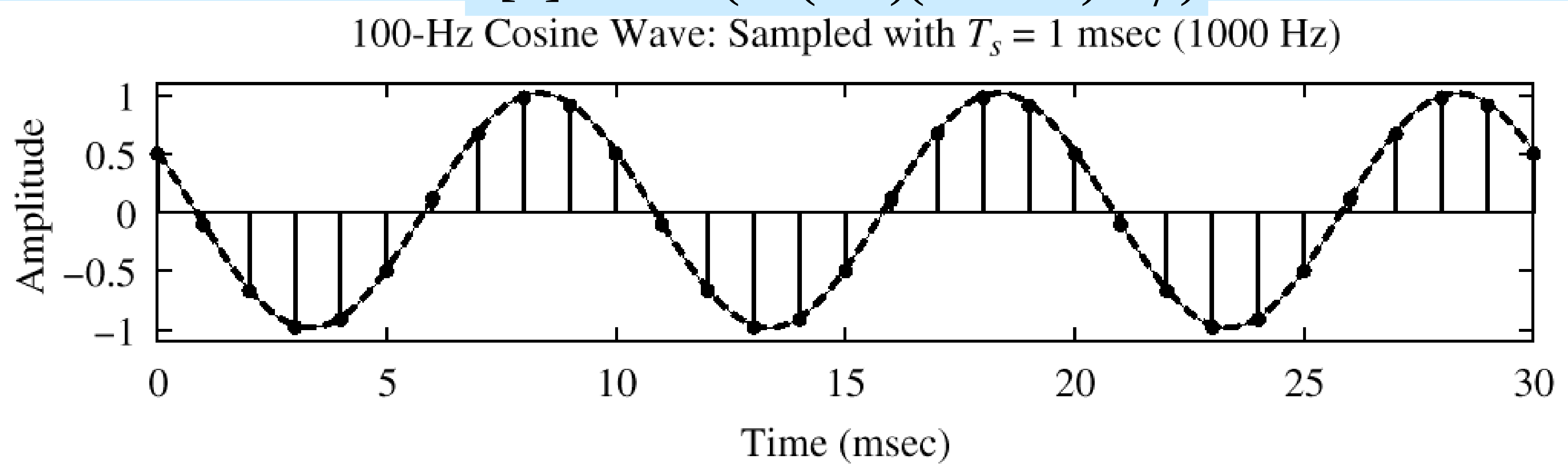
SPECTRUM (MORE LINES)

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

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



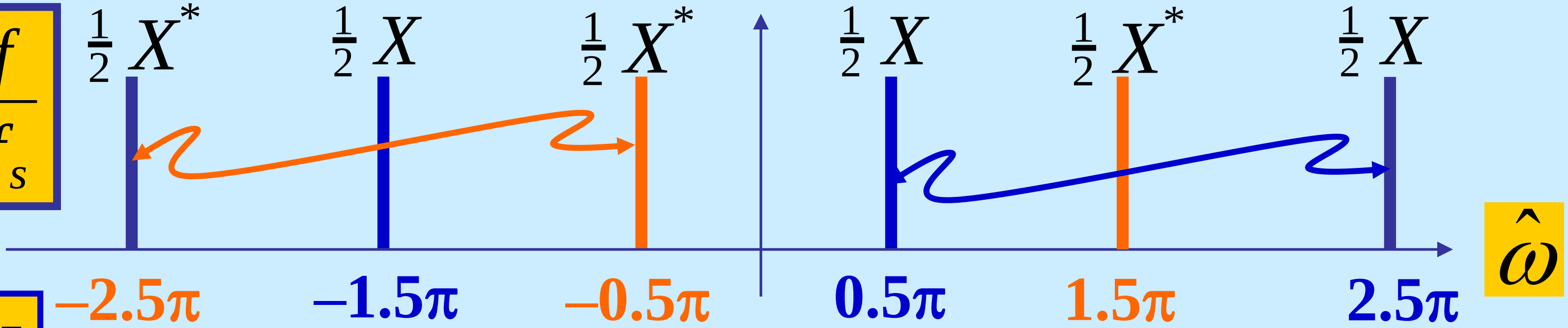
$$x[n] = A\cos(2\pi(100)(n/1000) + \varphi)$$



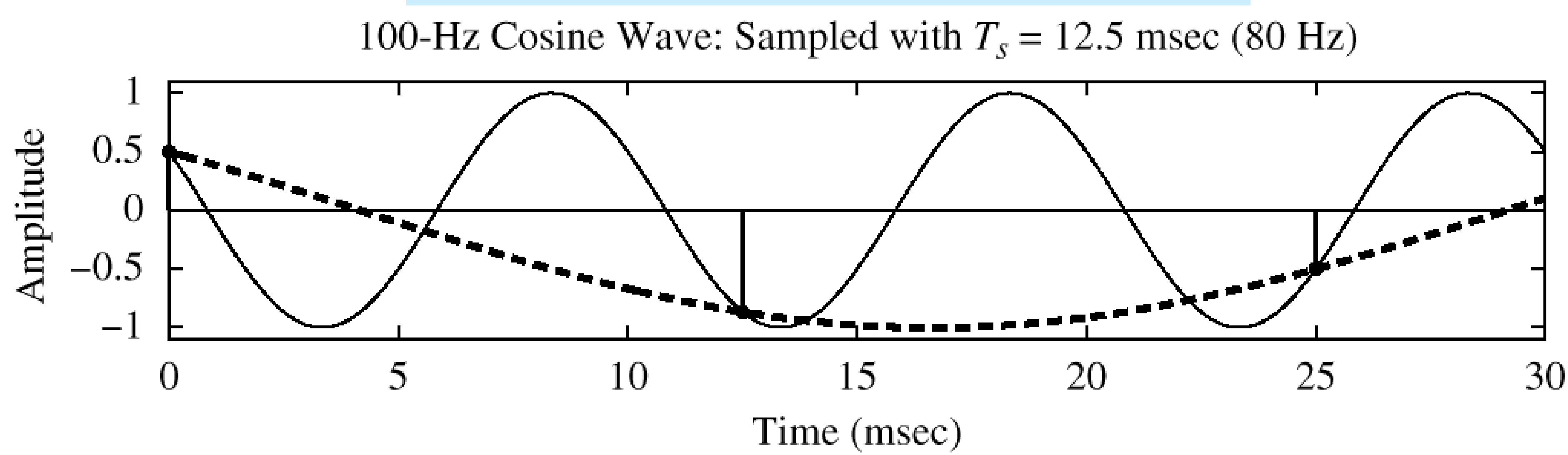
SPECTRUM (ALIASING CASE)

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

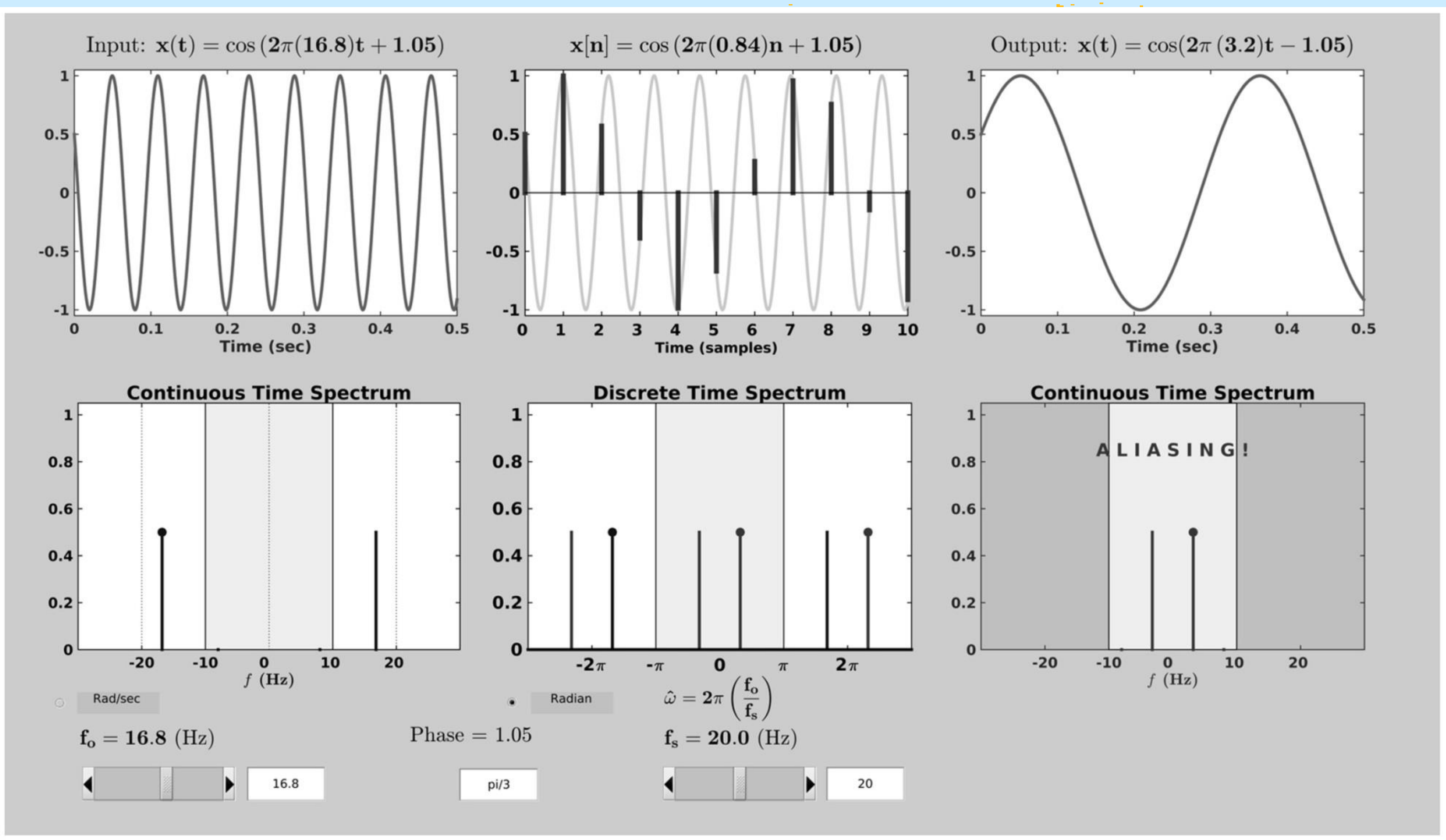
$$f_s = 80\,\text{kHz}$$



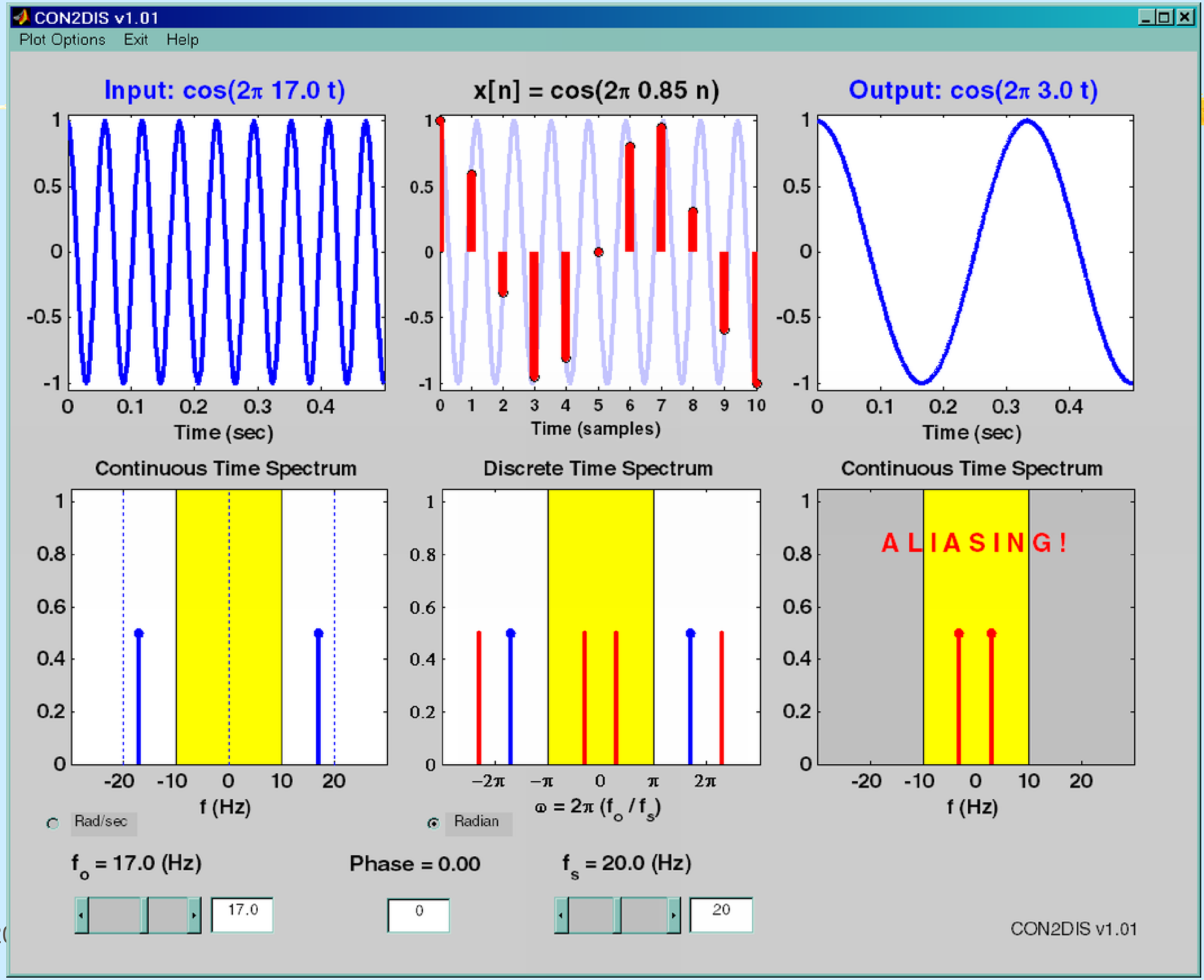
$$x[n] = A\cos(2\pi(100)(n/80) + \varphi)$$



SAMPLING GUI (con2dis)



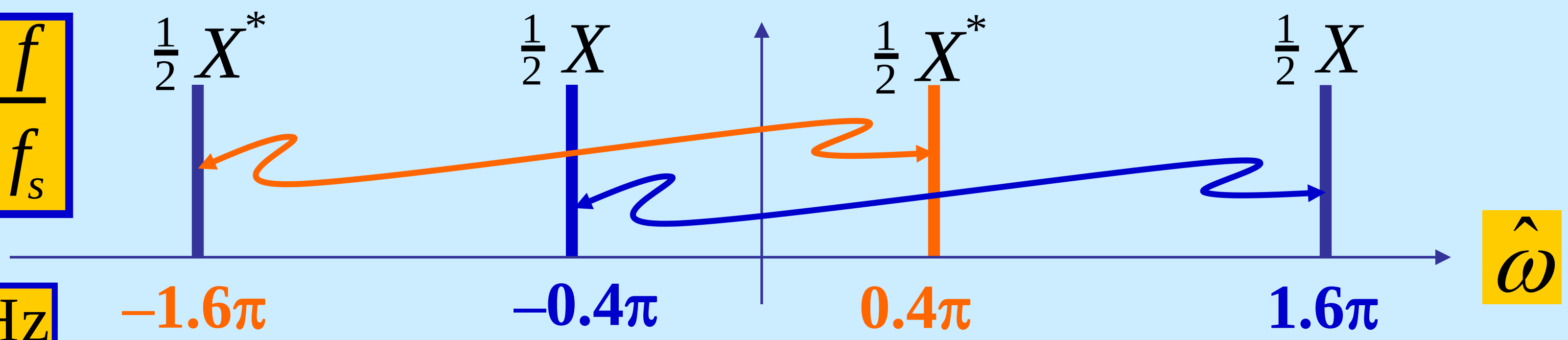
SAMPLING GUI (con2dis)



SPECTRUM (FOLDING CASE)

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

$$f_s = 125\,\text{Hz}$$



$$x[n] = A\cos(2\pi(100)(n/125) + \varphi)$$