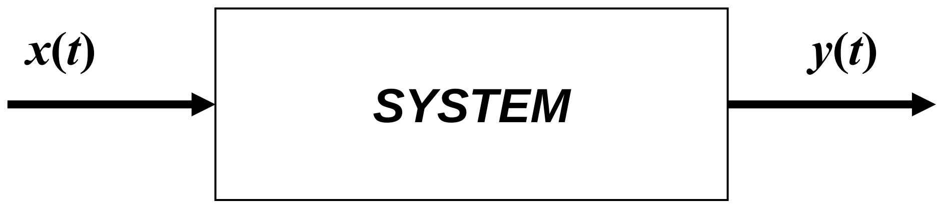
Chapter 4
Sampling and Aliasing

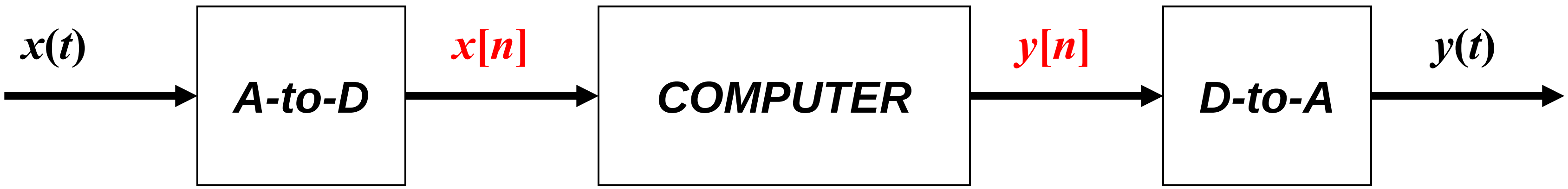
SYSTEMS Process Signals.



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

  - Extract information from $x(t)$.

System IMPLEMENTATION



- Discrete-time signal
  - The time variable is discrete.
- DIGITAL/MICROPROCESSOR
  - Digital signal : The amplitude is also discrete.
- 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.

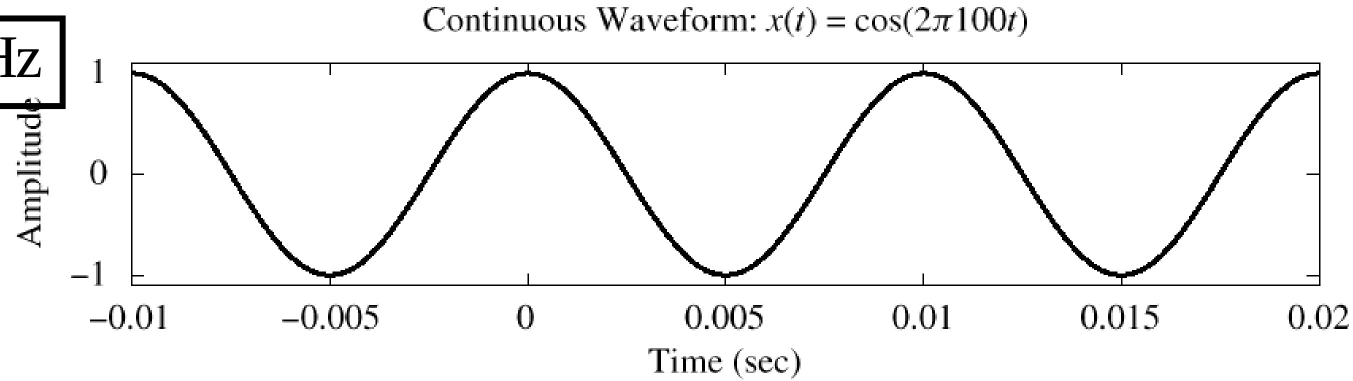
SAMPLING $x(t)$

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

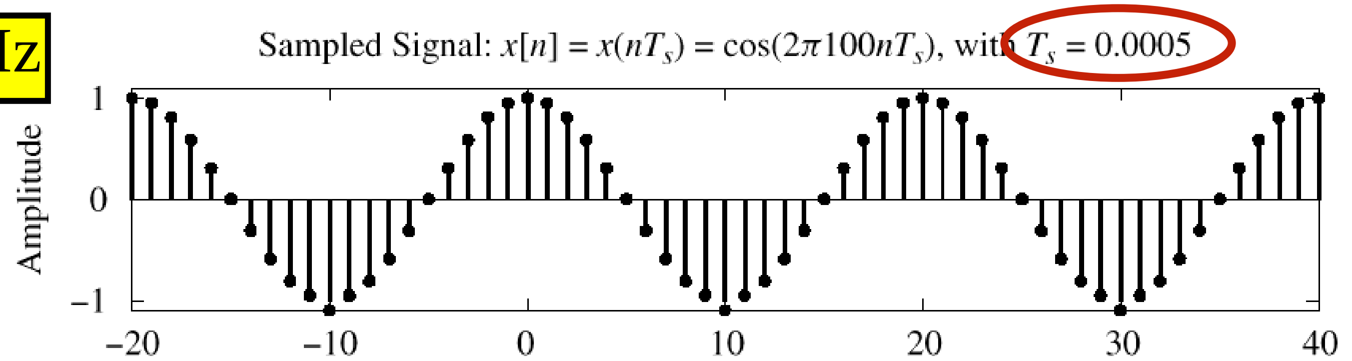
$$x(t) \longrightarrow \boxed{\textit{C-to-D}} \longrightarrow x[n]=x(nT_s)$$

SAMPLING RATE, f_s

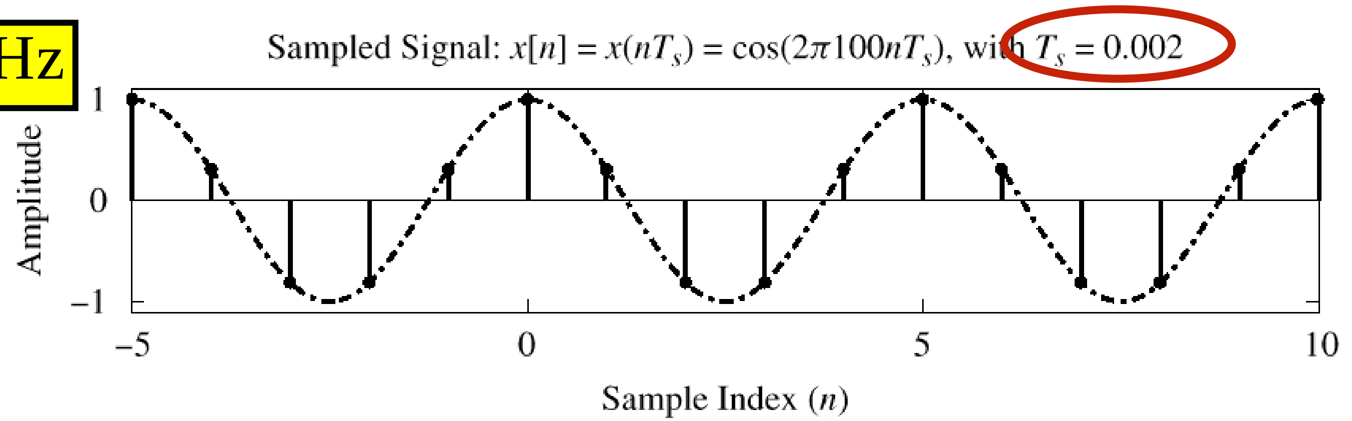
$f = 100\text{Hz}$



$f_s = 2\text{ kHz}$



$f_s = 500\text{Hz}$



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.
- The number of bytes for 1 minute is
  - 2 X (16/8) X 60 X 44100 = 10.584 Mbytes.

SAMPLING THEOREM

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

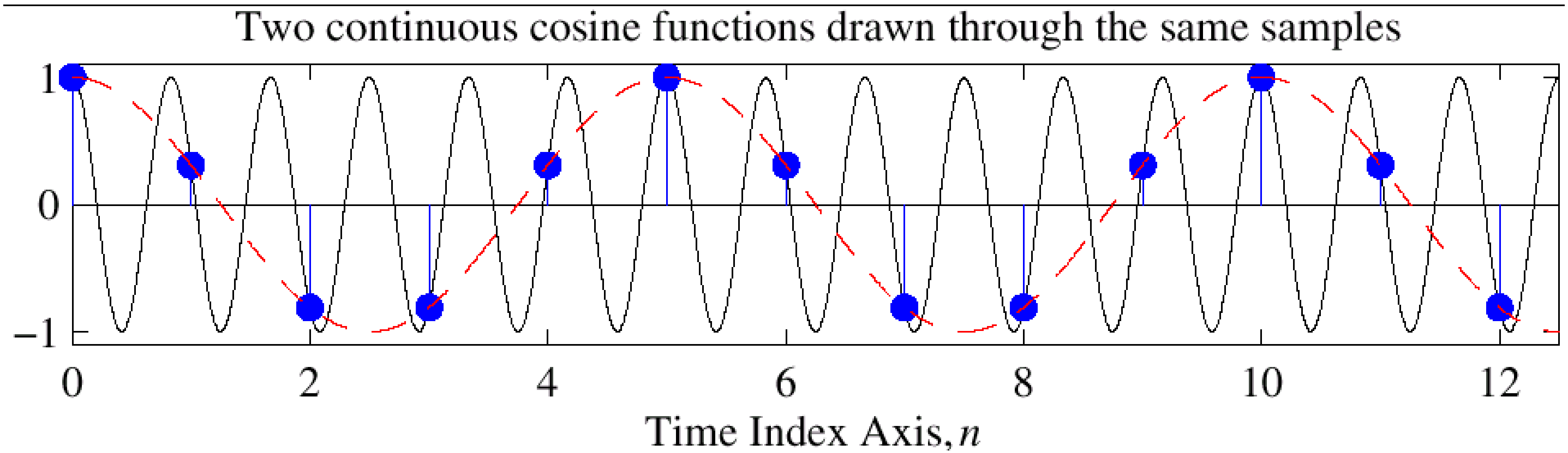
Shannon Sampling Theorem

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 $2f_{\max}$.

- Nyquist rate
  - The minimum sampling rate of $2f_{max}$

Reconstruction? Which One?

- Given the samples, draw a sinusoid through the values.



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

When n is an integer,

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

DISCRETE-TIME SINUSOID

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

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

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

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

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

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

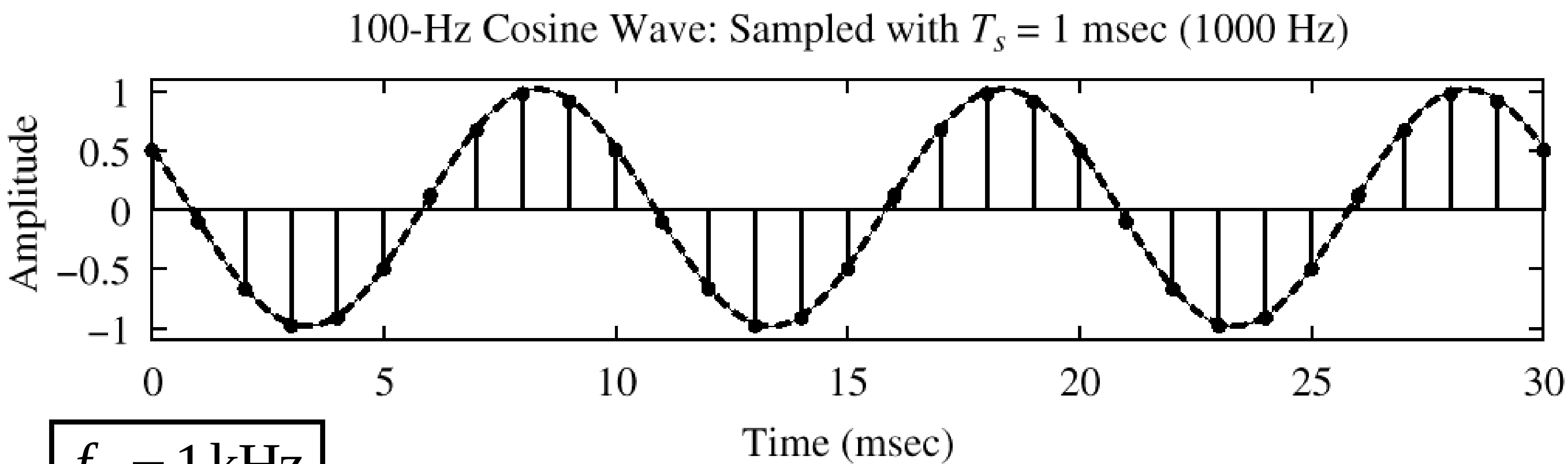
DEFINE DISCRETE-TIME(DIGITAL) FREQUENCY.

DISCRETE-TIME(DIGITAL) FREQUENCY $\hat{\omega}$

- $\hat{\omega}$ VARIES from **0** to **2π**, as *f* varies from 0 to the sampling frequency.
- Since the signal is assumed to contain only components with frequencies up to $f_s/2$, $\hat{\omega}$ varies from -π to π.
- UNITS are radians, **not** rad/sec.
    - DIGITAL FREQUENCY is NORMALIZED.

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

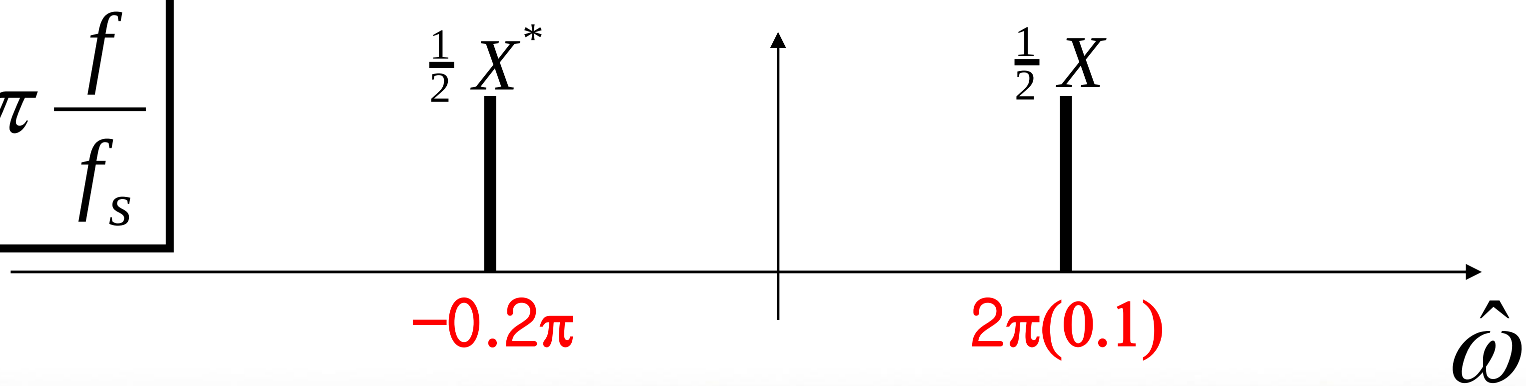
SPECTRUM (DIGITAL) (1)



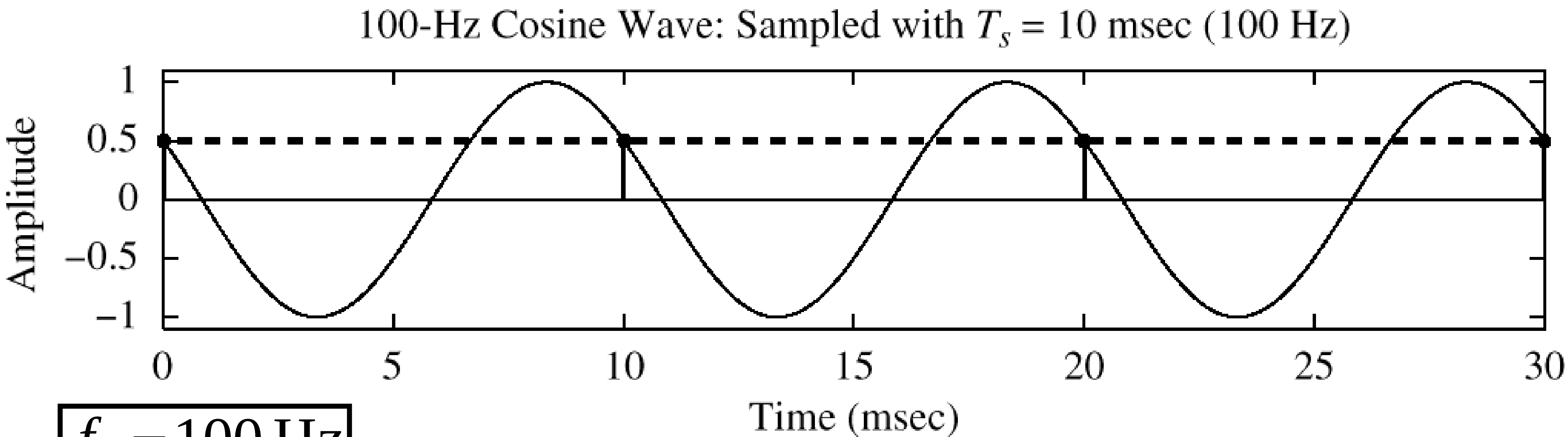
$$\boxed{f_s = 1\,\text{kHz}}$$

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

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



SPECTRUM (DIGITAL) (2)

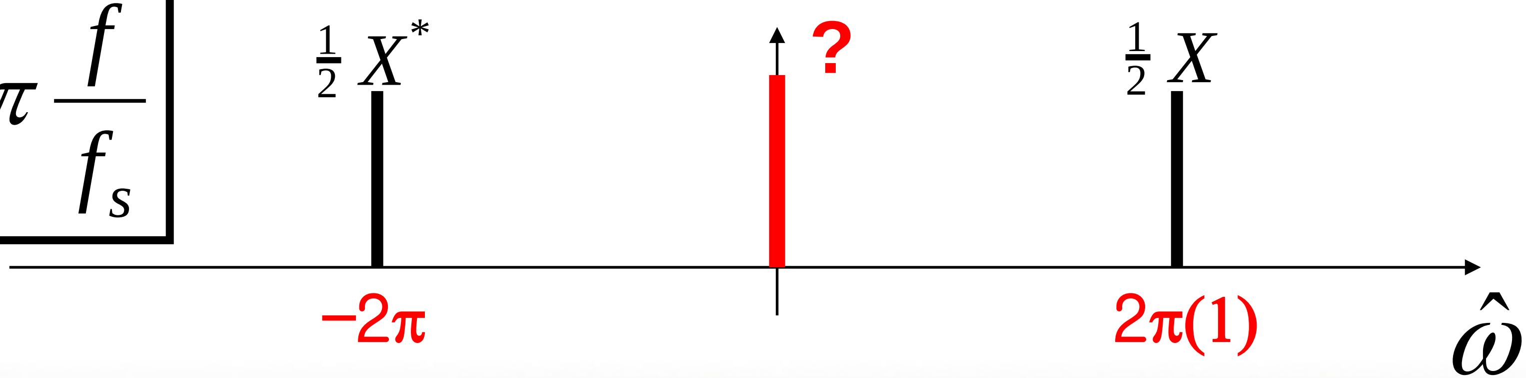


$$\boxed{f_s = 100\text{ Hz}}$$

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

$x[n]$ is zero frequency???

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



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 a period **2π** because

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

ALIASING DERIVATION (1)

- Other frequencies give the same $\hat{\omega}$.

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

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

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

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

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

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

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

ALIASING DERIVATION (2)

- Other frequencies give the same $\hat{\omega}$.

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

$$x(n/f_s) = A\cos(2\pi(f + \ell f_s)n/f_s + \varphi)$$

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

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

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

ALIASING CONCLUSIONS

- ADDING f_s, $2f_s$, 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 $(f_o + 2f_s)$.

NORMALIZED FREQUENCY

- DIGITAL FREQUENCY

Normalized Radian Frequency

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

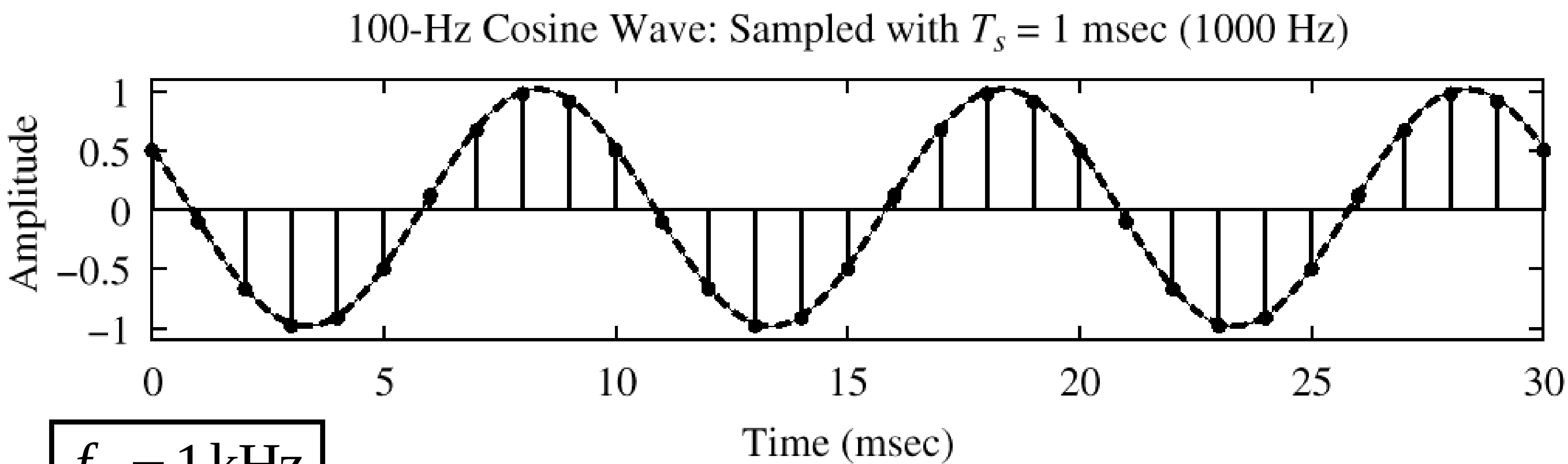
Normalized Cyclic Frequency

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

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.

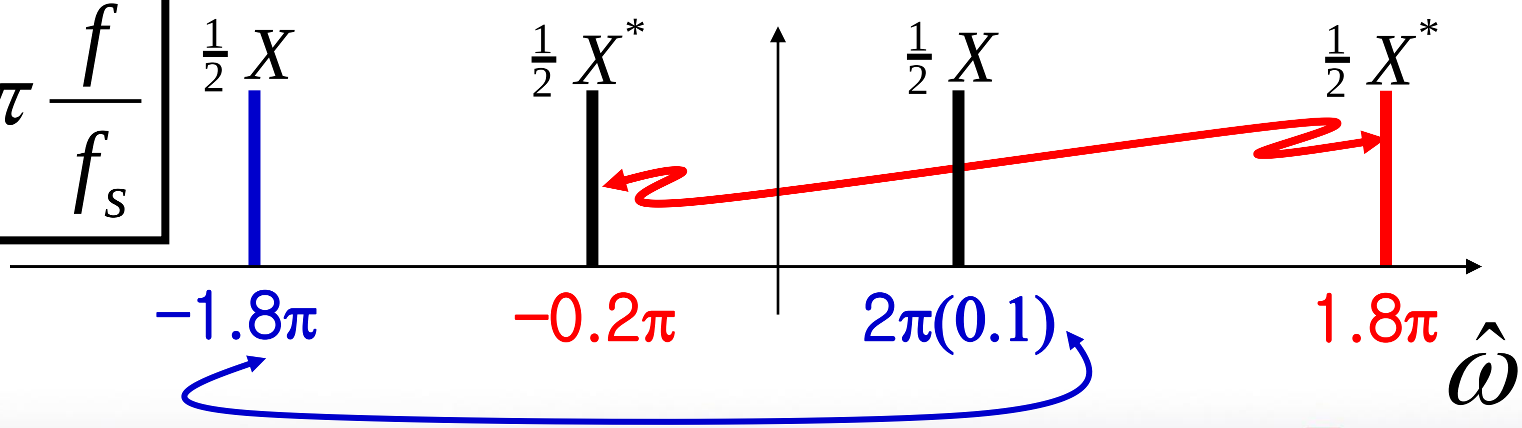
SPECTRUM (MORE LINES)



$$\boxed{f_s = 1\,\mathrm{kHz}}$$

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

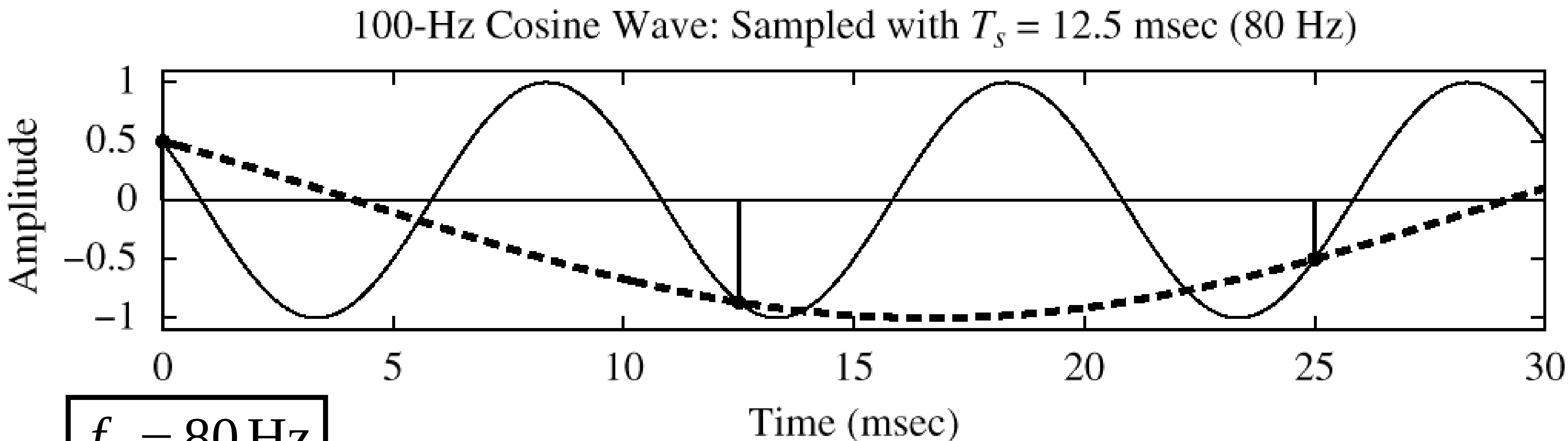
$$\boxed{\hat{\omega} = 2\pi \frac{f}{f_s}}$$



SPECTRUM (MORE LINES) (2)

- $x[n] = A\cos(0.2\pi n+\phi)$
- 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.:
  - $\{1.8\pi, 3.8\pi, 5.8\pi, \ldots\}$ & $\{-2.2\pi, -4.2\pi, \ldots\}$

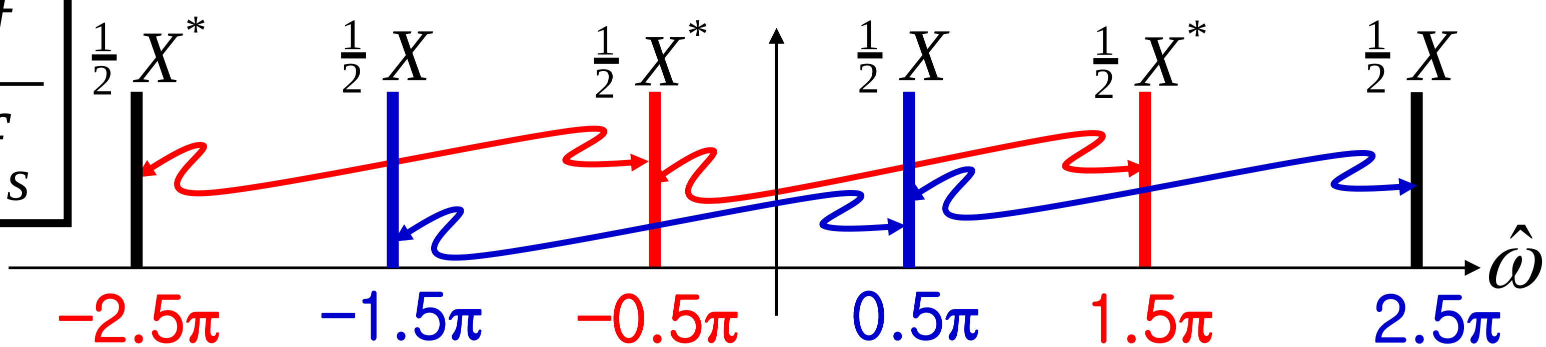
SPECTRUM (ALIASING CASE)



$$\boxed{f_s = 80\,\text{Hz}}$$

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

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



FOLDING DERIVATION

- Negative frequencies can give the same $\hat{\omega}$.

$$x(t) = A\cos(2\pi(f + \ell f_s)t + \varphi)$$

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

$$x[n] = x(nT_s) = A\cos(2\pi(-f + \ell f_s)nT_s - \varphi)$$

$$x[n] = A\cos((-2\pi f T_s)n + (2\pi\ell f_s T_s)n - \varphi)$$

$$x[n] = A\cos((2\pi f T_s)n - \underline{2\pi\ell n} + \varphi)$$ $\boxed{\cos(-\theta) = \cos\theta}$

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

SAME DIGITAL SIGNAL

FOLDING (a type of ALIASING)

- MANY $x(t)$ give IDENTICAL $x[n]$.
- CAN'T TELL f_o FROM (f_s-f_o), $(2f_s-f_o)$, or $(3f_s-f_o)$.
- EXAMPLE:
  - $y(t)$ has a 1000 Hz component.
  - SAMPLING FREQ = 1500 Hz
  - WHAT is the “FOLDED” ALIAS?

$$-1000 + 1500 \rightarrow 500$$

DIGITAL FREQUENCY

- DIGITAL FREQUENCY $\hat{\omega}$

Normalized Radian Frequency

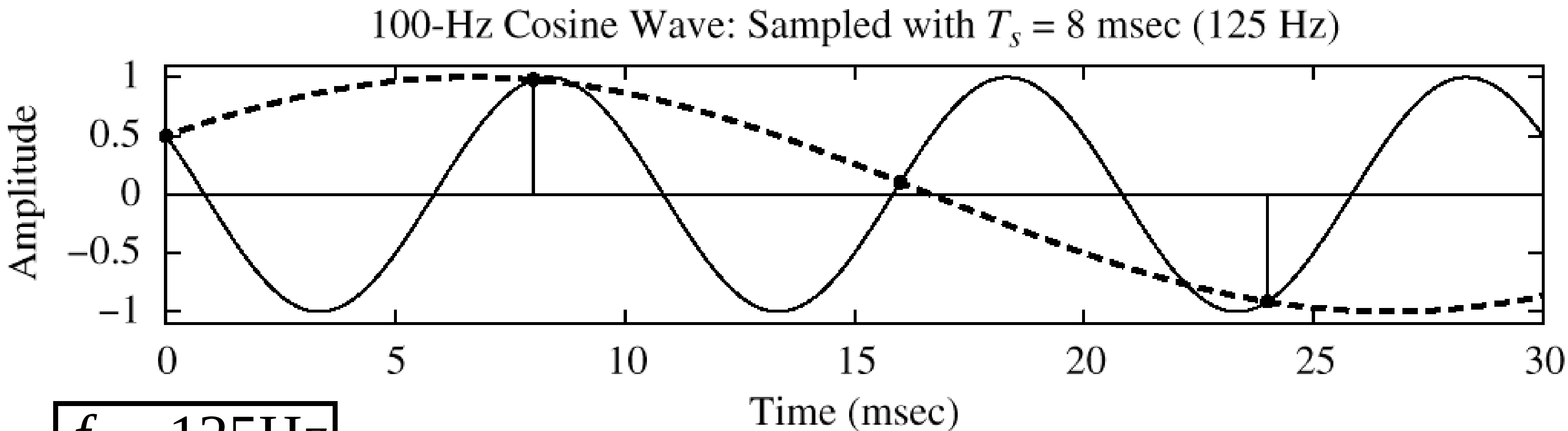
$$\boxed{\hat{\omega} = \omega T_s = \frac{2\pi f}{f_s}} + 2\pi\ell$$

ALIASING

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

FOLDED ALIAS

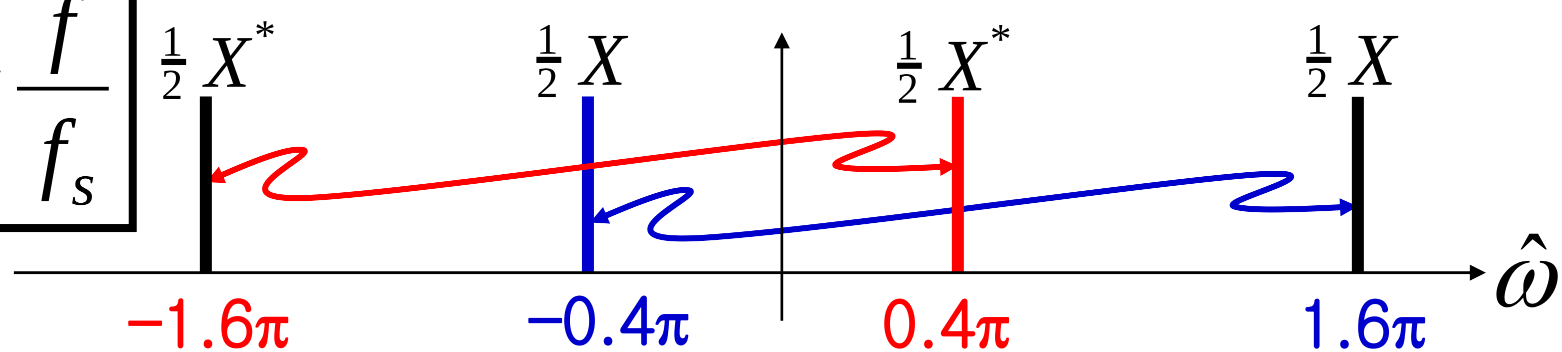
SPECTRUM (FOLDING CASE)



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

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

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



FOLDING DIAGRAM

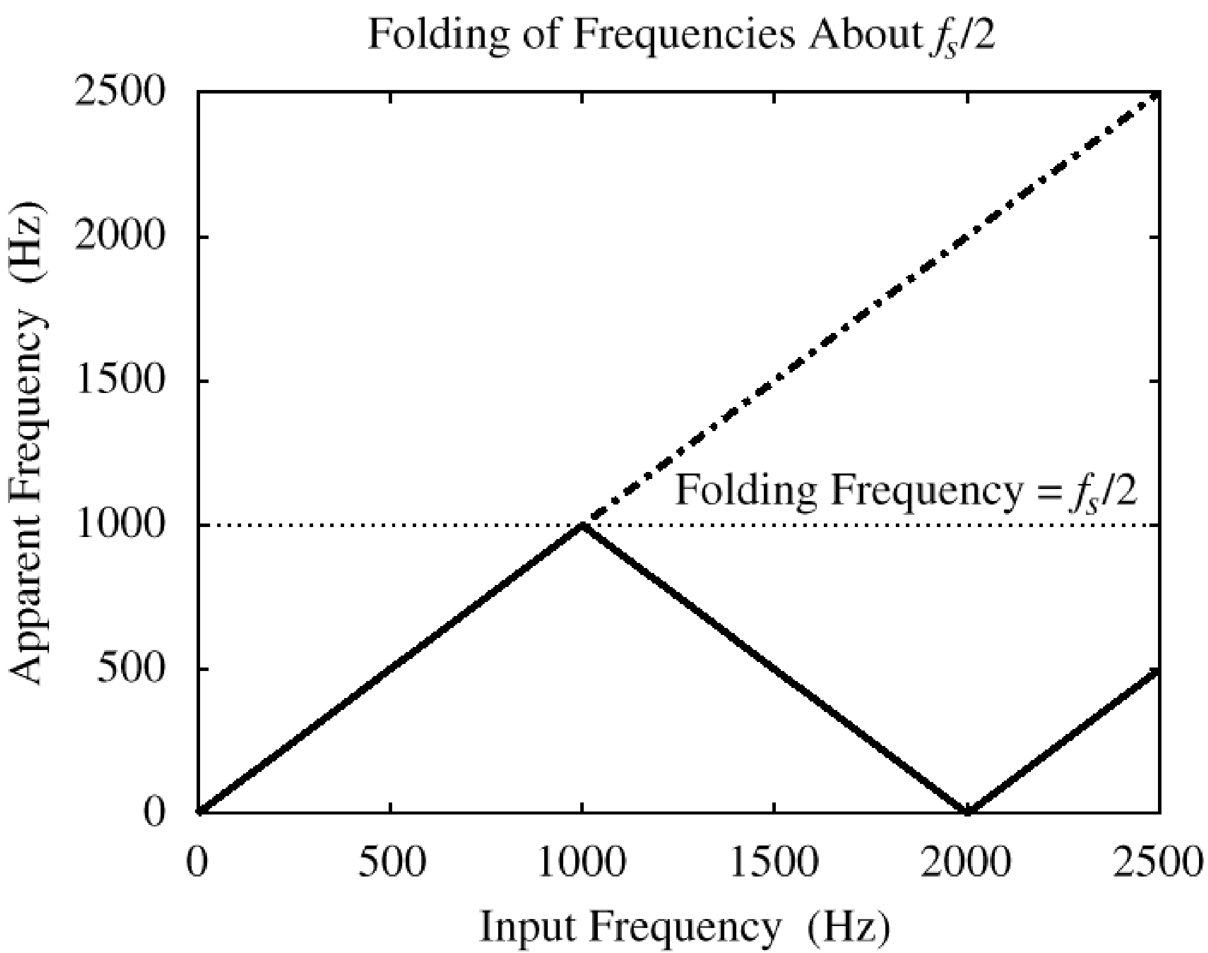
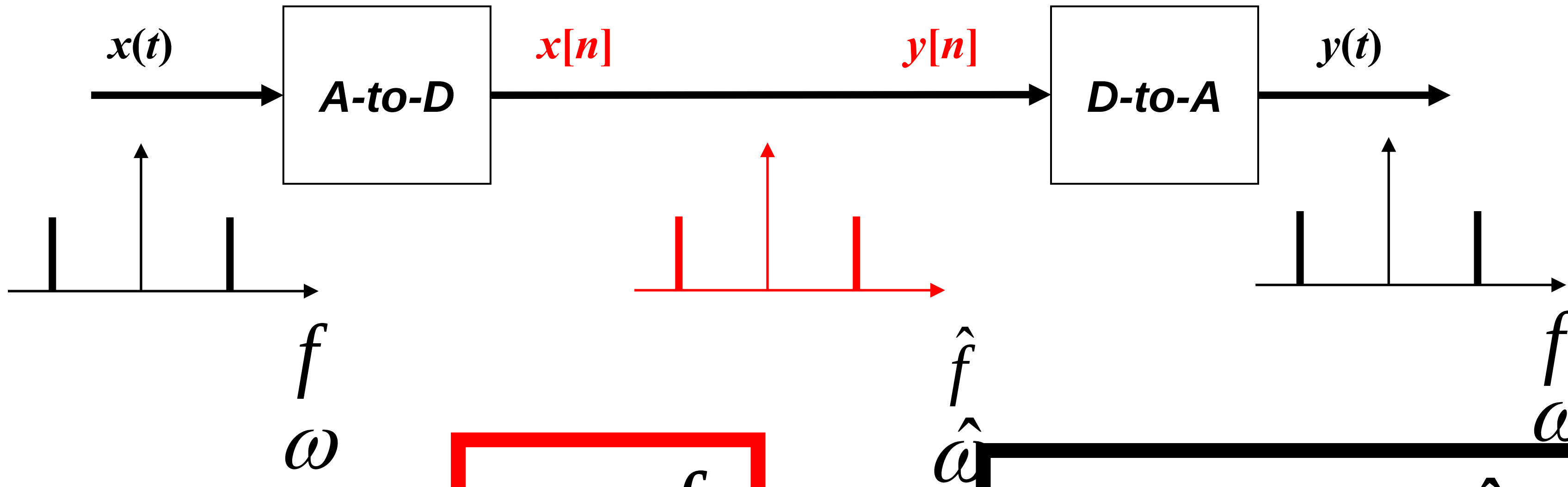


Figure 4-12 Folding of a sinusoid sampled at $f_s = 2000$ samples/sec. The apparent frequency is the lowest frequency of a sinusoid that has exactly the same samples as the input sinusoid.

ALIASING & FOLDING

- $x(t)$ = SINUSOID @ f_o
- SAMPLED SIGNAL: $x[n] = x(n/ f_s)$
- **ALIASING**:
  - $x[n]$ COULD BE FROM $(f_o + f_s)$, $(f_o - f_s)$, $(f_o + 2f_s)$, $(f_o - 2f_s)$, etc.
- **FOLDING**:
  - A type of ALIASING
  - x[n] COULD BE FROM $(-f_o + f_s)$, $(-f_o - f_s)$, $(-f_o + 2f_s)$, $(-f_o - 2f_s)$, et c.

FREQUENCY DOMAINS



$$\boxed{\hat{f} = \frac{f}{f_s}} + \ell$$

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

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

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

D-to-A Reconstruction

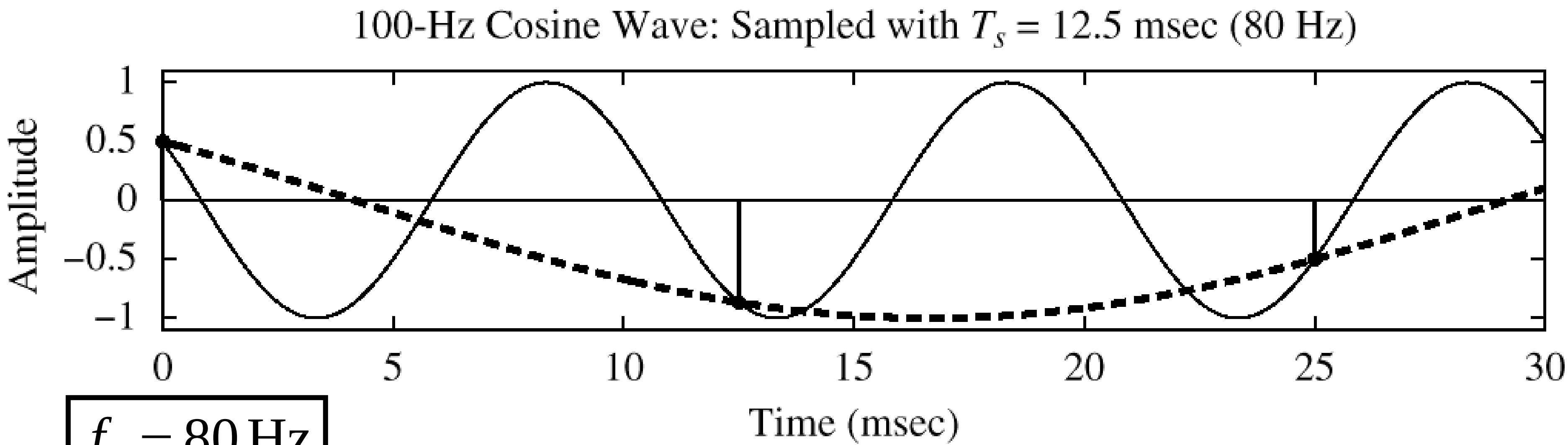
$$x(t) \longrightarrow \boxed{\text{A-to-D}} \xrightarrow{x[n]} \boxed{\text{COMPUTER}} \xrightarrow{y[n]} \boxed{\text{D-to-A}} \xrightarrow{y(t)}$$

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

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 $y[n]$ is a sinusoid.

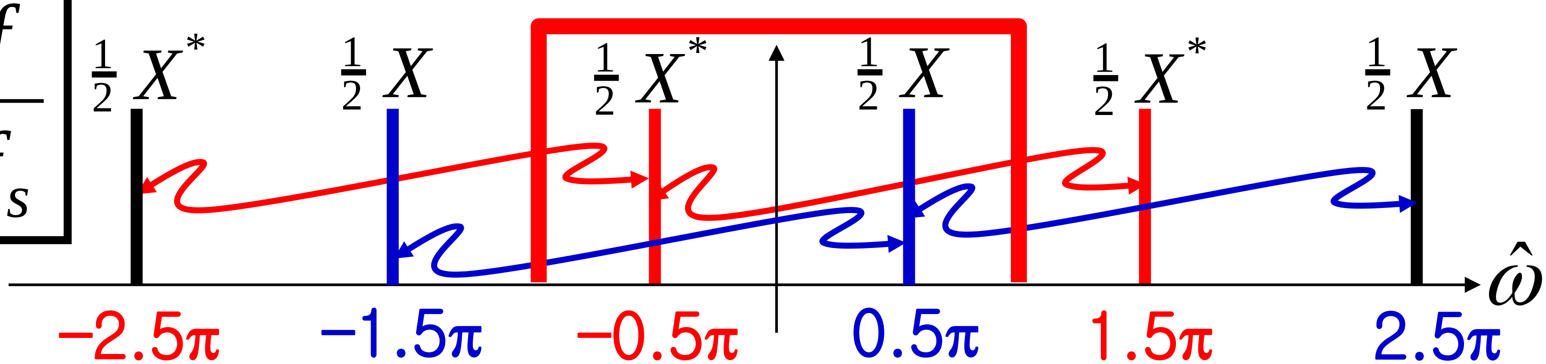
SPECTRUM (ALIASING CASE)



$$\boxed{f_s = 80\,\text{Hz}}$$

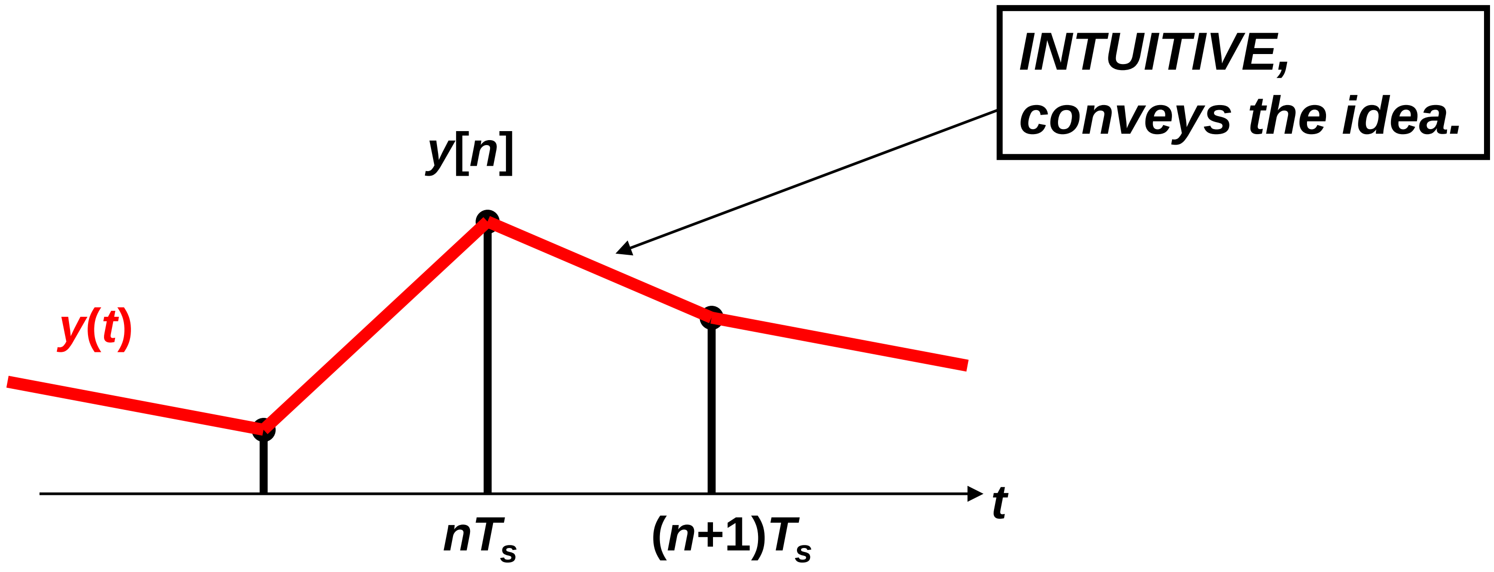
$$x[n] = A\cos(2\pi(100)(n/80) + \varphi)$$

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



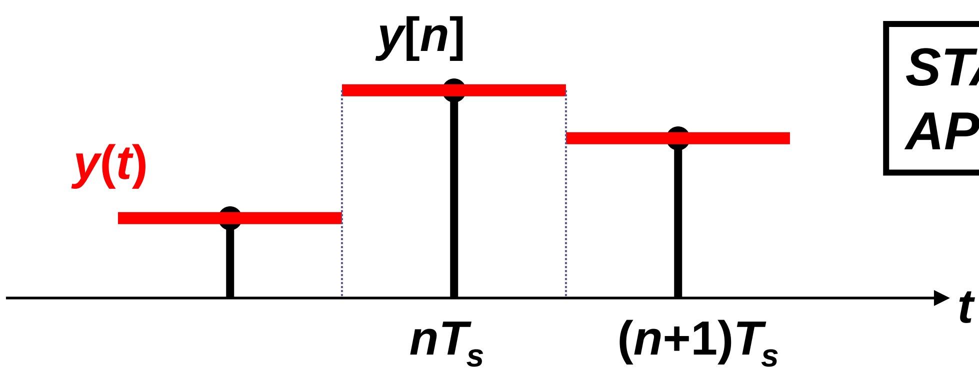
Reconstruction (D-to-A)

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



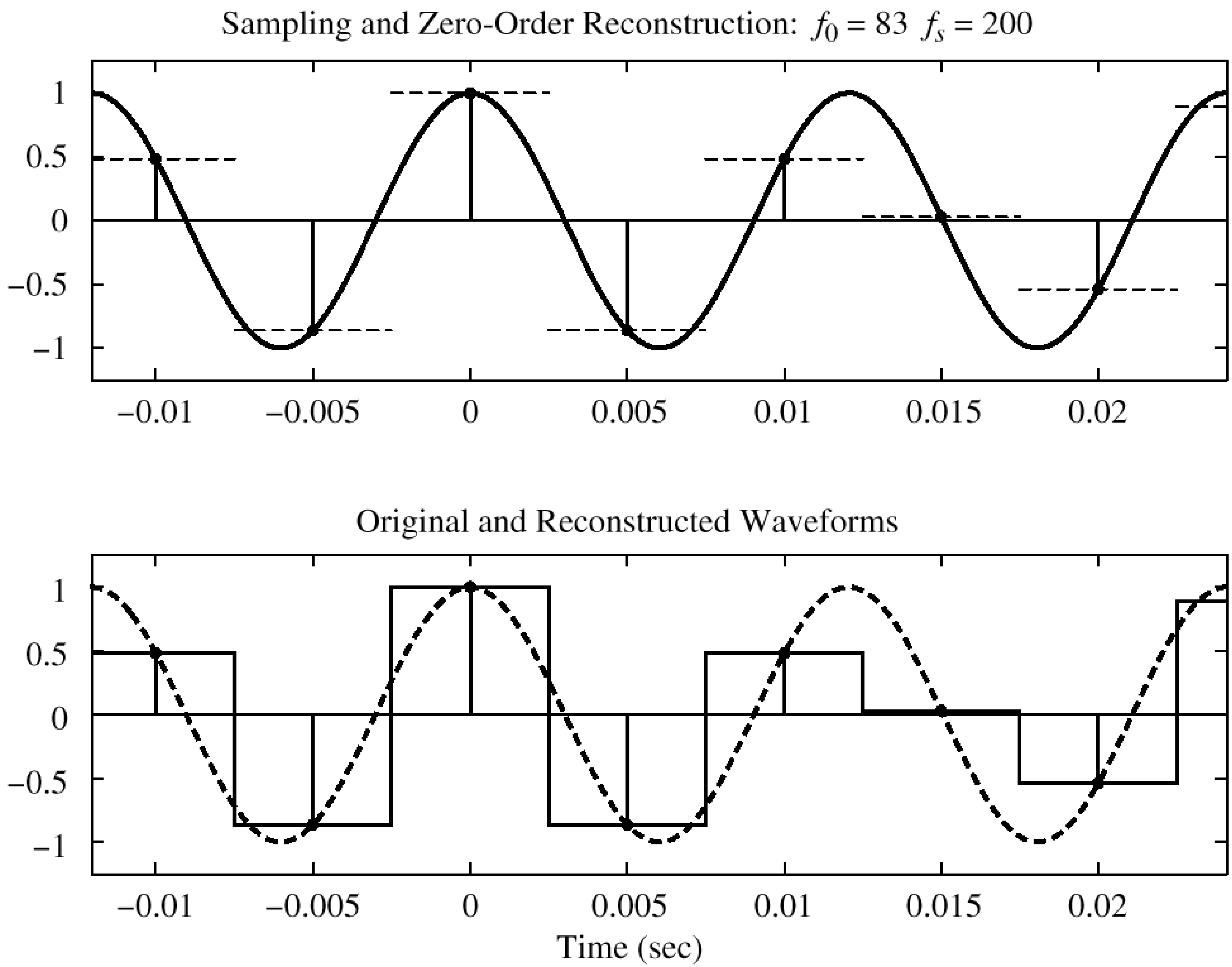
SAMPLE & HOLD DEVICE

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

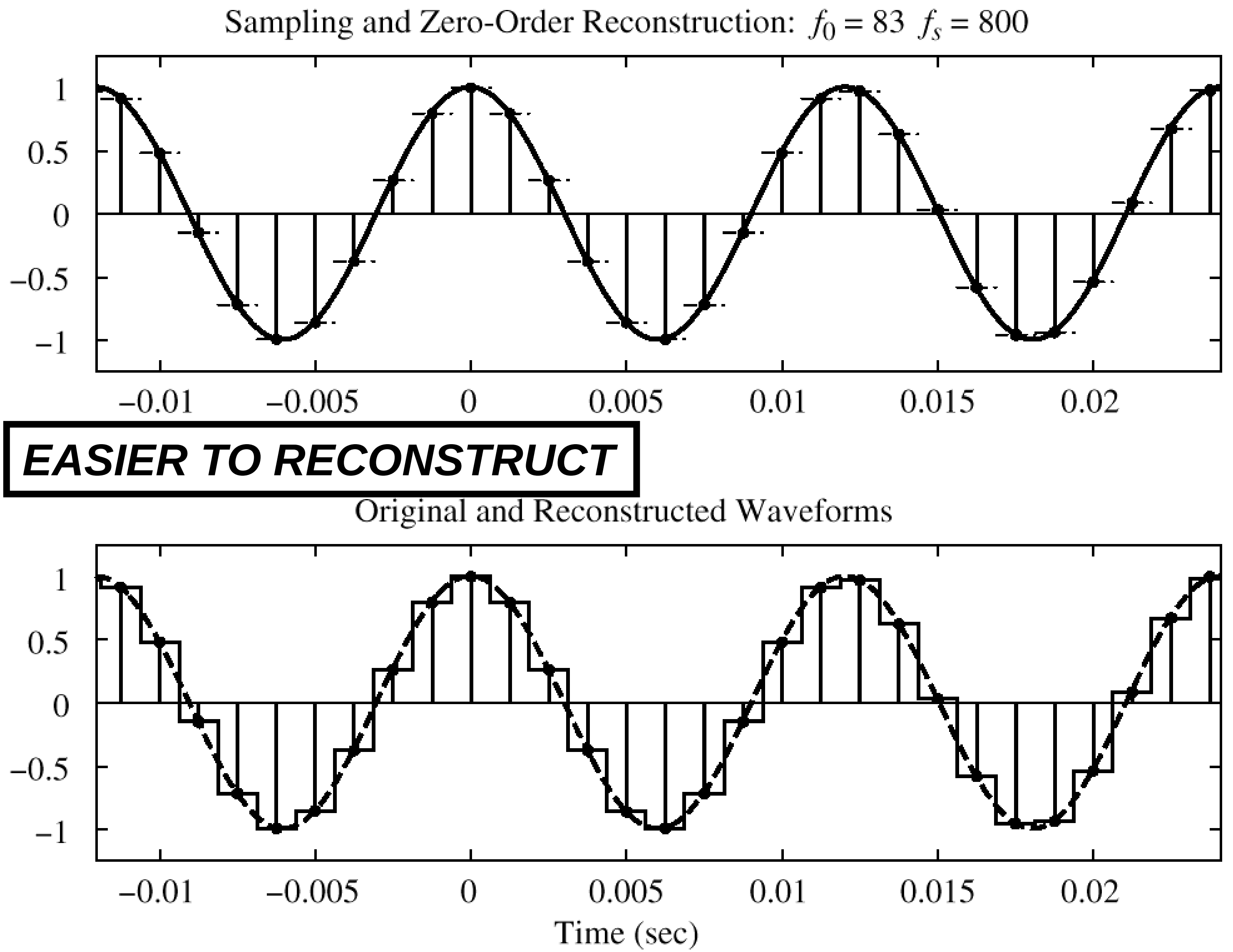


STAIR-STEP APPROXIMATION

SQUARE PULSE CASE



OVER-SAMPLING CASE



MATH MODEL for D-to-A

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

- SQUARE PULSE:

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

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 in a "REAL TIME".

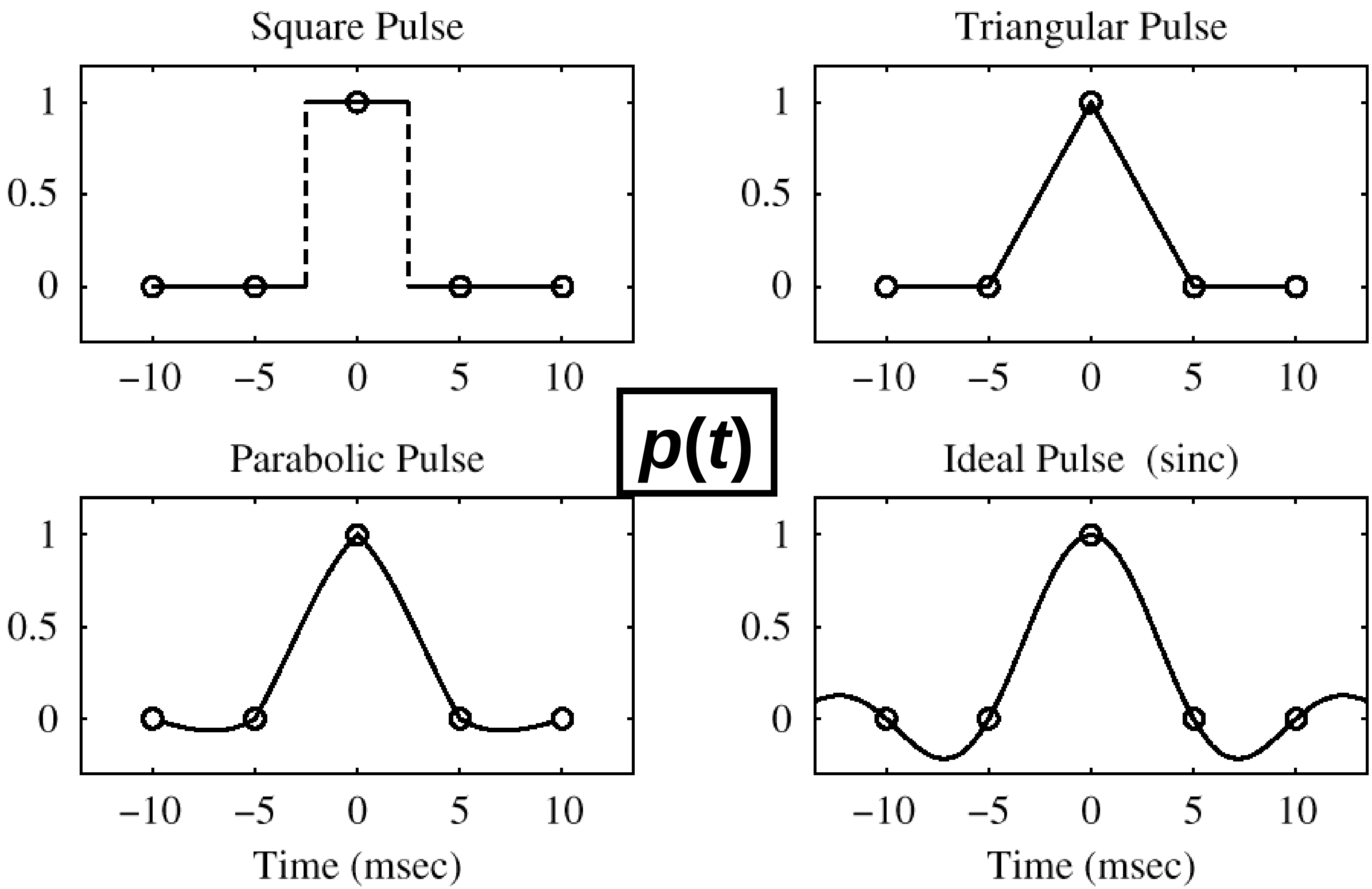
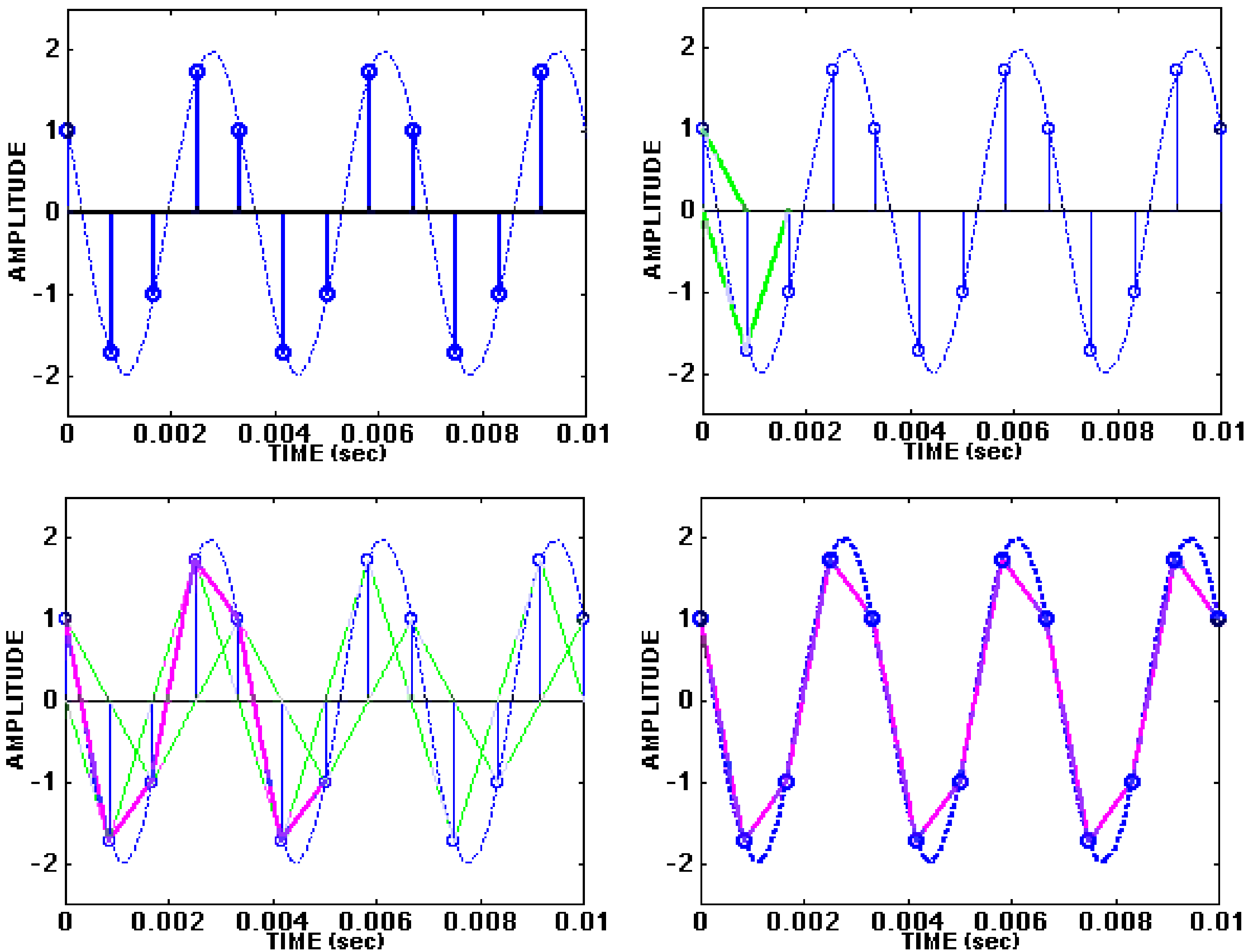


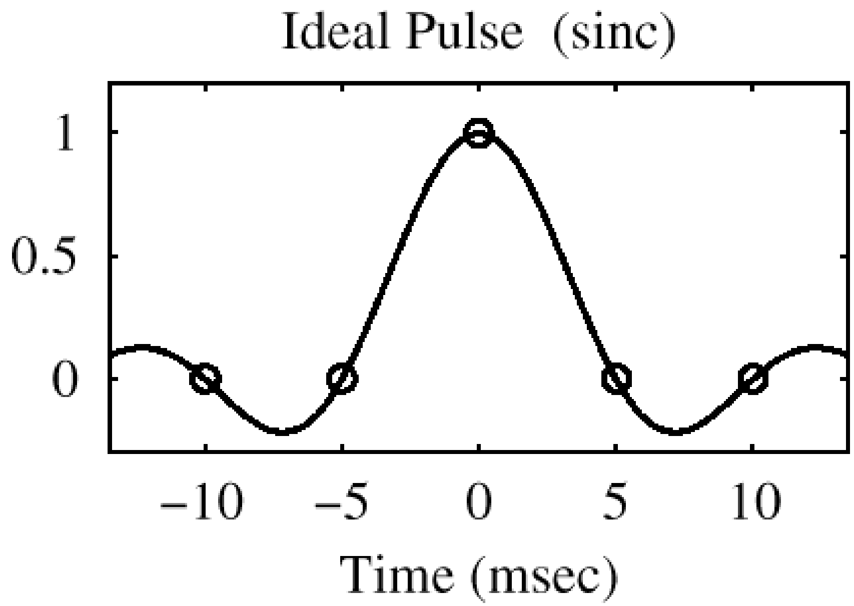
Figure 4.17 Four different pulses for D-to-C conversion. The sampling period is $T_s = 0.005$, i.e., $f_s = 200$ Hz. Note that the duration of each pulse is approximately one or two times T_s.

TRIANGULAR PULSE (2 x the Nyquist rate)



OPTIMAL PULSE?

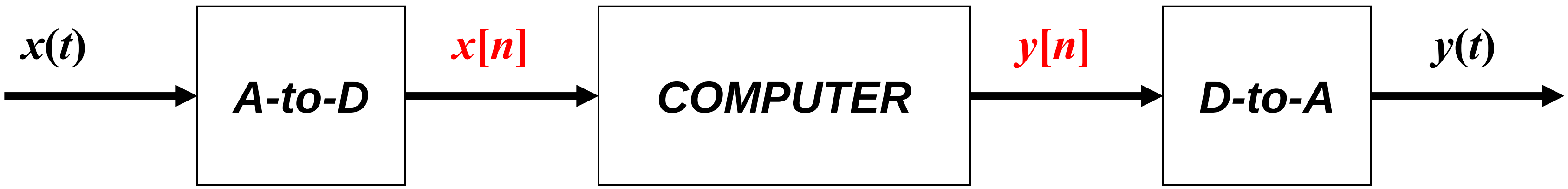
CALLED "BANDLIMITED INTERPOLATION"



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

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

D-to-A Reconstruction



- Create continuous $y(t)$ from $y[n]$.
  - **REALISTIC CONSTRAINT:** SMOOTH $y(t)$.
    - Use the lowest possible frequency.
  - $y[n]$ is a list of numbers.
  - How fast?
  - In MATLAB: `soundsc(yy,fs)`

Thank you

- Homework
  - P-4.1, 3, 4, 5, 6, 11, 12, 13, 17, & 18
- Reading assignment
  - Section 5-1, 2, & 3