# ENGR 071
# Digital Signal Processing

Class 05

02/04/2025

# ENGR 71 Class 05

- Class Overview
  - Review Assignment 2
  - Sampling
  - Introduce some Matlab tools for signal analysis and filtering

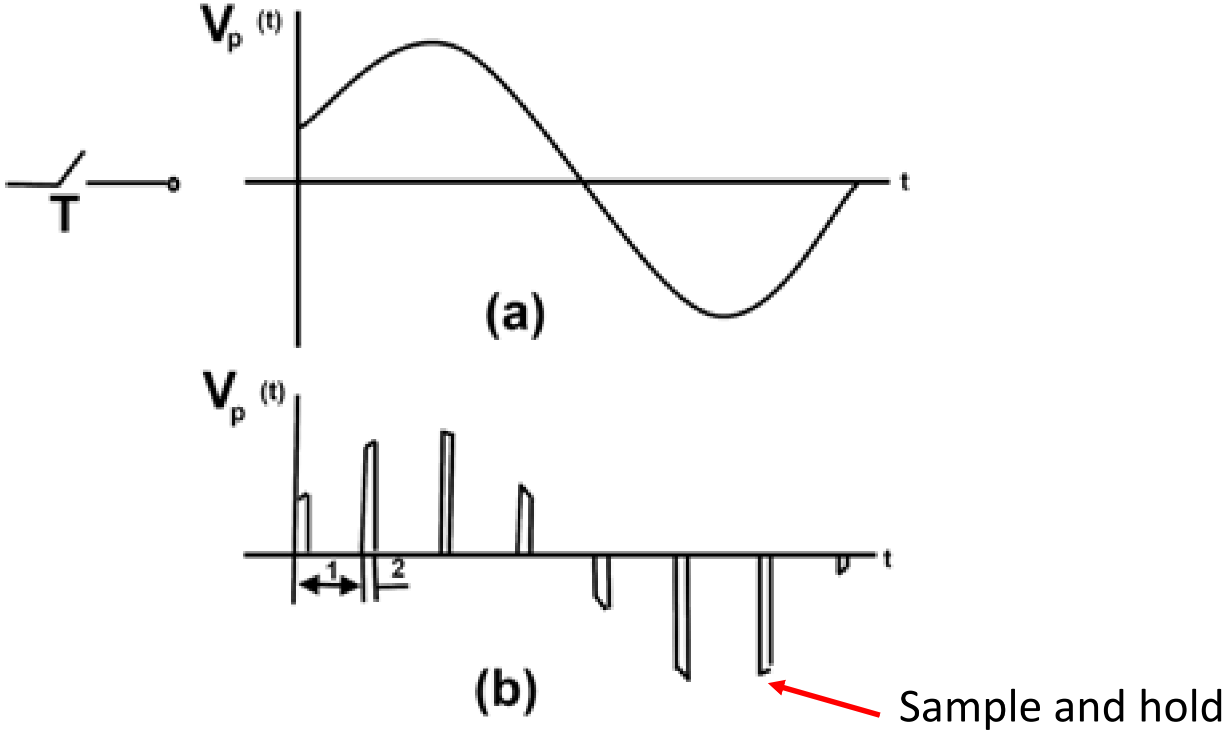
# SAMPLING
# CONTINUOUS → DISCRETE SIGNALS

# Sampling for Discrete Signals

- Discrete Signals
  - Fundamental issue: How do you pick the sampling interval?
    - Time steps too small – redundant data
    - Time steps too large – data loss
    - You can figure out the optimal step size based on the frequency content of the data.
  - How do you get a discrete signal?
    - Take samples of continuous signal at fixed time-steps
    - Sampling pulse must have some width, but usually, we neglect this width

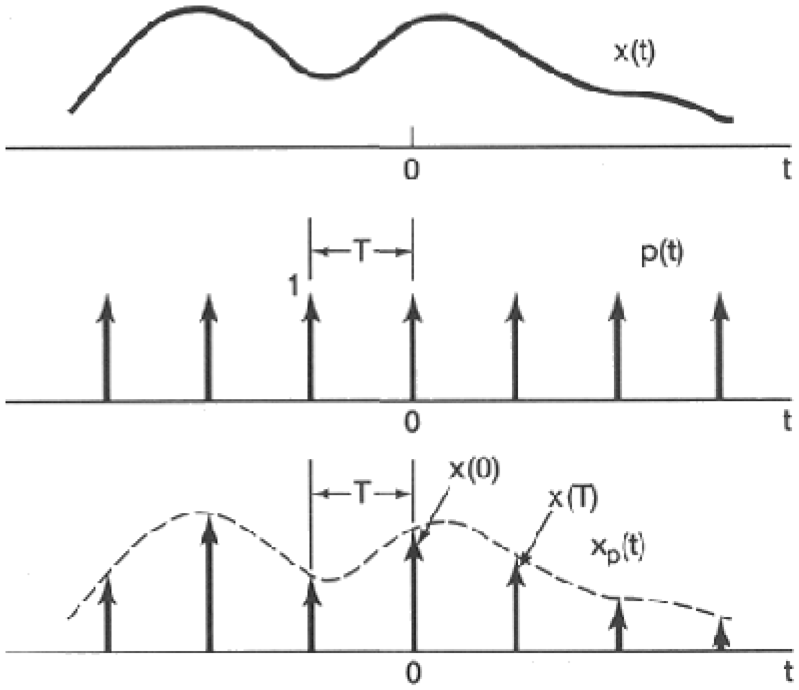
# Sampling for Discrete Signals

- Sampling (showing width of sampling pulse)

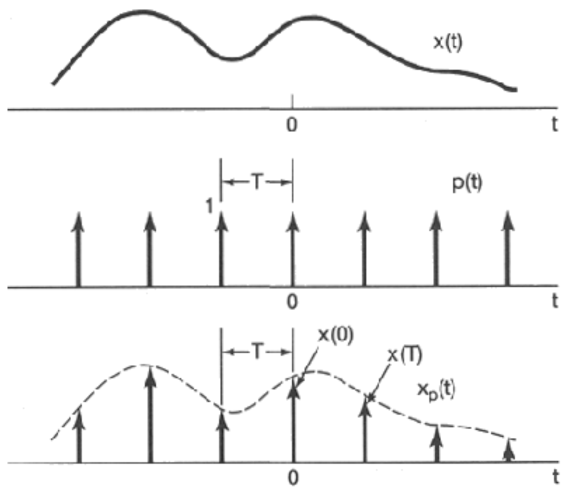


# Sampling for Discrete Signals

- Sampling (ignore width of sampling pulse)



# Sampling for Discrete Signals




- Sampling a signal at sample interval $T_s$
  - Signal: $x(t)$
  - Sampled signal: $x_s(t)$
  - Sampling function: $\delta_{T_s}(t) = \sum_n \delta(t - nT_s)$

    $\delta(t)$ is unit impulse
  - Multiply signal by the sampling function

$$x_s(t) = x(t)\delta_{T_s}(t) = \sum_n x(t)\delta(t - nT_s) = \sum_n x(nT_s)\delta(t - nT_s)$$

- The last sum is true since only the values at $t = nT_s$ matters.
- **Notice both sampled signal and original signal are shown as continuous functions of time.**
- The discrete signal could be represented as a sequence of numbers:

$$\{x_k\} = \{\cdots x(-kT_s), \cdots, x(-2T_s), x(-1T_s), x(0T_s), x(1T_s), x(2T_s), \cdots, x(kT_s), \cdots\}$$

# Sampling for Discrete Signals

- Aliasing and the Nyquist-Shannon sampling theorem:
  - The sampling function is periodic,

    $\delta_{Ts}(t)$ periodic with a period of $T_s$
  - Since it is periodic, we can find its Fourier series:

$$f(t) \equiv \delta_{T_s}(t) \text{ with period } T_s \qquad \left( \text{Fundamental frequency } \Omega_s = \frac{2\pi}{T_s} \right)$$

(Here, I will use a capital Omega to represent the sampling frequency)

Definition of Fourier Series

$$f(t) = \sum_{n=-\infty}^{+\infty} D_n e^{jn\omega_0 t}$$

$$D_n = \frac{1}{T_0} \int_{T_0} f(t) e^{-jn\omega_0 t} dt$$

# Sampling for Discrete Signals

$$f(t) \equiv \delta_{Ts}(t) \ \text{ with period } T_s \qquad \left( \text{Fundamental frequency } \ \Omega_s = \frac{2\pi}{T_s} \right)$$

$$D_n = \frac{1}{T_s} \int_{-T_s/2}^{+T_s/2} \delta(t) e^{-jn\,\Omega_s\,t} dt = \frac{1}{T_s} e^{-jn\Omega_s 0} = \frac{1}{T_s}$$

$$f(t) = \sum_{n=-\infty}^{+\infty} D_n e^{jn\omega_0 t} \quad \Rightarrow \quad \delta_{T_s}(t) = \sum_{n=-\infty}^{+\infty} \frac{1}{T_s} e^{jn\Omega_s t} = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} e^{jn\Omega_s t}$$

– Recall that the sampled signal is: $x_s(t) = x(t)\delta_{T_s}(t)$

$$x_s(t) = x(t)\delta_{T_s}(t) = x(t)\frac{1}{T_s}\sum_{n=-\infty}^{+\infty} e^{jn\Omega_s t} = \frac{1}{T_s}\sum_{n=-\infty}^{+\infty} x(t)e^{jn\Omega_s t}$$

# Sampling for Discrete Signals

– Big question: What is the frequency content of the sampled signal and how does it compare to the frequency content of the original signal?
– Take the Fourier transform of both.

The Fourier transform of the original signal is:

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

# Sampling for Discrete Signals

– The Fourier transform of the sampled signal is:

$$X_s(\omega) = \int_{-\infty}^{\infty} x_s(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} x(t) e^{jn\Omega_s t} e^{-j\omega t} dt$$

$$= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) e^{jn\Omega_s t} e^{-j\omega t} dt = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) e^{-j(\omega - n\Omega_s)t} dt$$

$$X_s(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\Omega_s)$$

# Sampling for Discrete Signals

– Interesting!

– Original signal has frequency "content"

$$X(\omega)$$

– Sampled signal has frequency "content"

$$X_s(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\Omega_s)$$

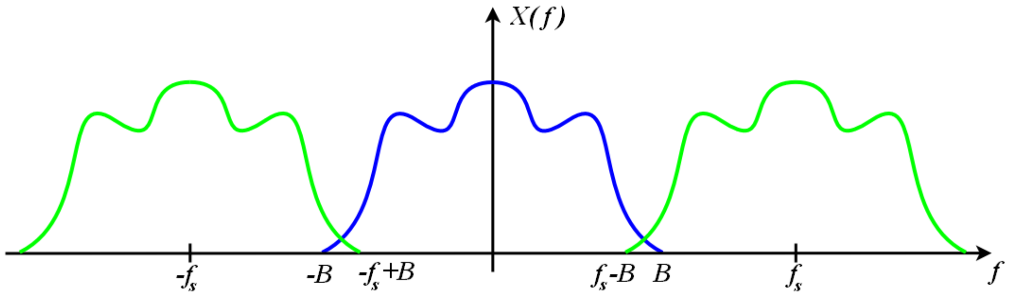
- Sampled signal has repeated copies of original frequency spectrum offset by $n\Omega_s$

– What does this mean?

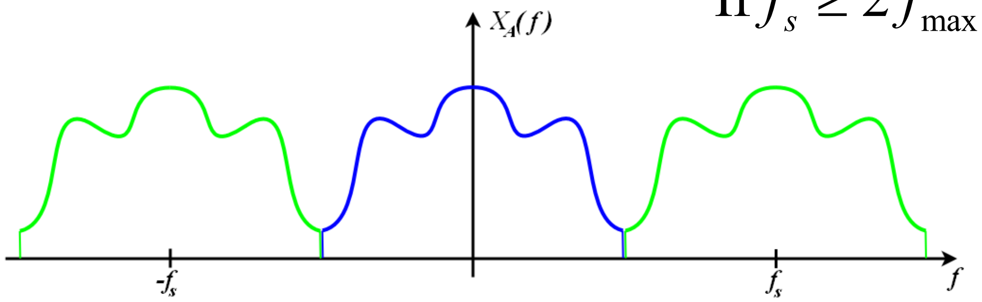
- Sampled signal will have frequencies that original doesn't have!
- High frequency components of signal may be overlapped by low frequency components of sampled signal

# Sampling for Discrete Signals

- Aliasing: Frequencies that are in sampled signal but not in original are called “aliased”

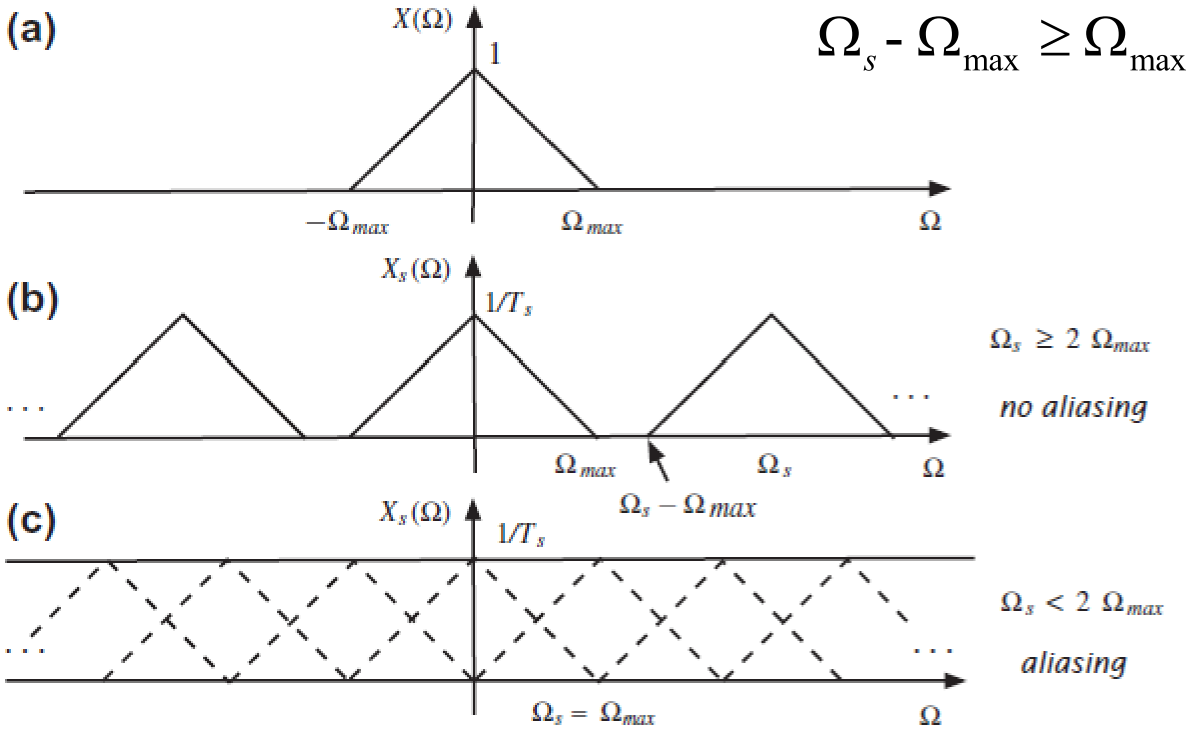



If $f_s \geq 2f_{\max}$, spectra do not overlap



# Sampling for Discrete Signals

- Aliasing: Frequencies that are in sampled signal but not in original are called "aliased"




spectra do not overlap if

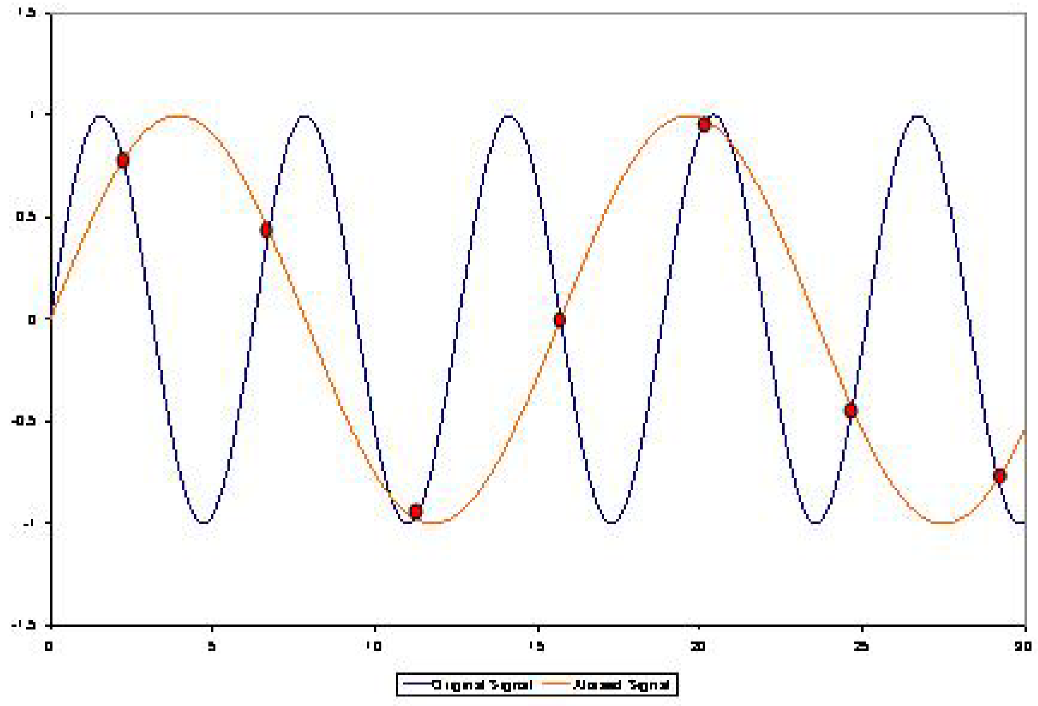
$$\Omega_s - \Omega_{\max} \geq \Omega_{\max} \quad \Rightarrow \quad \Omega_s \geq 2\Omega_{\max}$$

$$f_s \geq 2 f_{\max}$$

Nyquist sampling rate is minimum rate at which signal can be sampled to accurately reconstruct it from its samples

# Sampling for Discrete Signals

- Not that complicated:

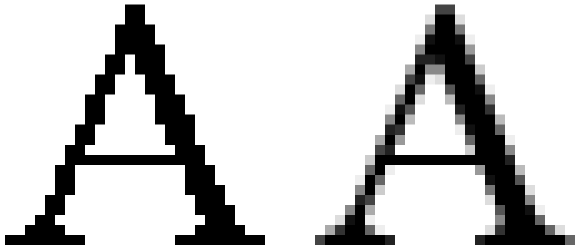


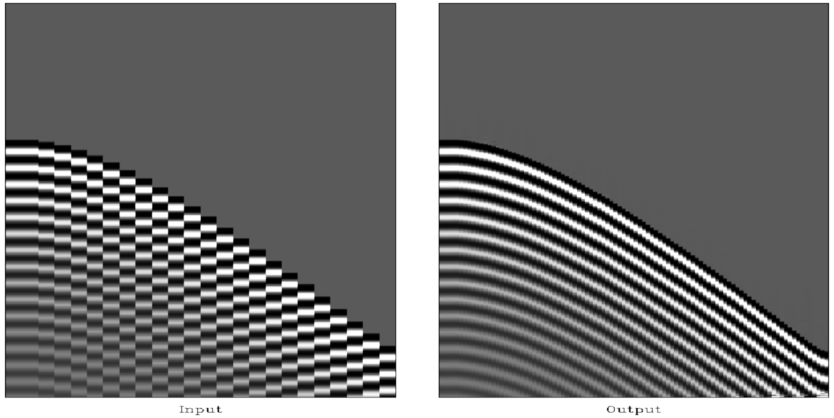
# Sampling for Discrete Signals

- Lots of interesting examples
  - A strobe light “samples” motion,
    jerky movements captured when light flashes
  - Fan blades that appear to turn backwards
  - Moire patterns are another example

# Sampling for Discrete Signals

– “jaggies”





# Sampling for Discrete Signals

- **Nyquist-Shannon sampling theorem**:
  – For a band-limited signal, there will be no overlapping frequencies if you sample at a rate twice the maximum frequency in the signal.

$$\boxed{\Omega_s \geq 2\omega_{\max} \quad \text{or} \quad F_s \geq 2f_{\max} \quad \text{or} \quad T_s \leq \frac{1}{2f_{\max}}}$$

# Sampling for Discrete Signals

- Consider digital music (CD's, Pandora, Spotify, …)
  - Sampling rate is 44.1 kHz
  - Good for frequencies up to 22.05 kHz which is about the upper limit of normal human hearing. (64 – 23,000 Hz)
  - Not so great for dogs (67-45,000) and cats (45-64,000)
  - Rotten for bats, (110 kHz), beluga whales (123 kHz) and porpoises (150 kHz)

# Sampling for Discrete Signals

- Reconstruction of original signal from sampled signal
  - If the original signal is band-limited, and $-\omega_{\max} \le \omega \le \omega_{\max}$
  - If sampled signal has sampling frequency $\Omega_s \ge 2\omega_{\max}$
  - Able to exactly recover the original signal from the sampled signal

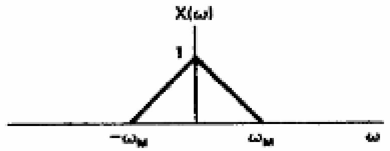
Frequency spectrum of original signal: $X(\omega)$

Frequency spectrum of sampled signal: $X_s(\omega) = \dfrac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\Omega_s)$

# Sampling for Discrete Signals

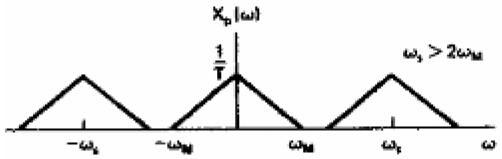
- Reconstruction of original signal from sampled signal

Original signal



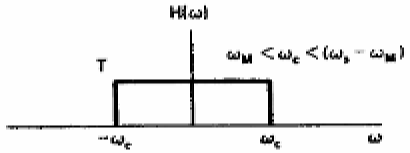
$$X(\omega)$$

Sampled signal
$\omega_s > 2\ \omega_m$



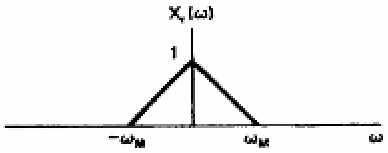
$$X_s(\omega) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} X(\omega - n\Omega_s)$$

Ideal
reconstruction filter
(low-pass)



$$H_{\text{rect}}(\omega) = T_s \quad -\Omega_s/2 \le \omega \le \Omega_s/2$$

Reconstructed signal
(=Original signal)



$$X_r(\omega) = X_s(\omega) H_{\text{rect}}(\omega)$$

# Sampling for Discrete Signals

- Reconstruction of original signal from sampled signal

$$X_s(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\Omega_s)$$

$$H_{\text{rect}}(\omega) = T_s \quad -\Omega_s/2 \le \omega \le \Omega_s/2$$

$$X_r(\omega) = X_s(\omega) H_{\text{rect}}(\omega)$$

- In the time domain: (Product in frequency-domain is convolution in time-domain)

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

$$h_{\text{sinc}}(t) = \frac{\sin(\pi t/T_s)}{(\pi t/T_s)}$$

$$x_r(t) = [x_s * h_{\text{sinc}}](t) = \int_{-\infty}^{\infty} x_s(\tau) h_{\text{sinc}}(t-\tau)\, d\tau$$

# Sampling for Discrete Signals

- Reconstruction of original signal from sampled signal

$$x_r(t) = [x_s * h_{\text{sinc}}](t) = \int_{-\infty}^{\infty} x_s(\tau) h_{\text{sinc}}(t-\tau)\, d\tau$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT_s)\delta(\tau - nT_s) h_{\text{sinc}}(t-\tau)\, d\tau$$

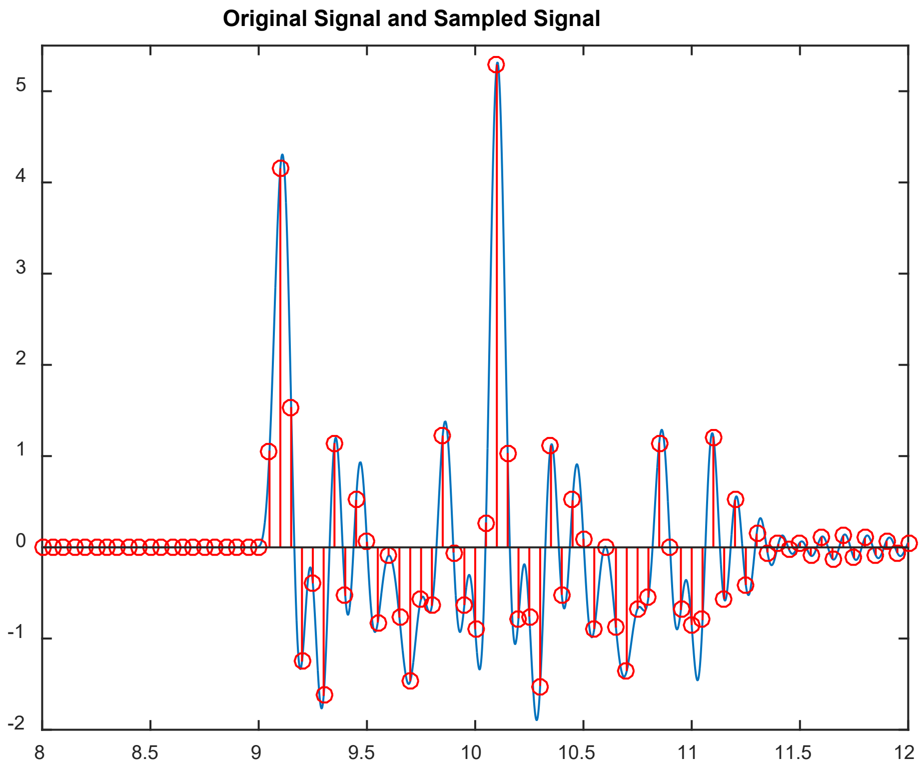
$$= \sum_{n=-\infty}^{\infty} x(nT_s) \int_{-\infty}^{\infty} \delta(\tau - nT_s) h_{\text{sinc}}(t-\tau)\, d\tau$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) h_{\text{sinc}}(t - nT_s)$$

$$h_{\text{sinc}}(t) = \frac{\sin(\pi t/T_s)}{(\pi t/T_s)}$$

$$\boxed{x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\sin\left(\pi (t - nT_s)/T_s\right)}{\left(\pi (t - nT_s)/T_s\right)}}$$

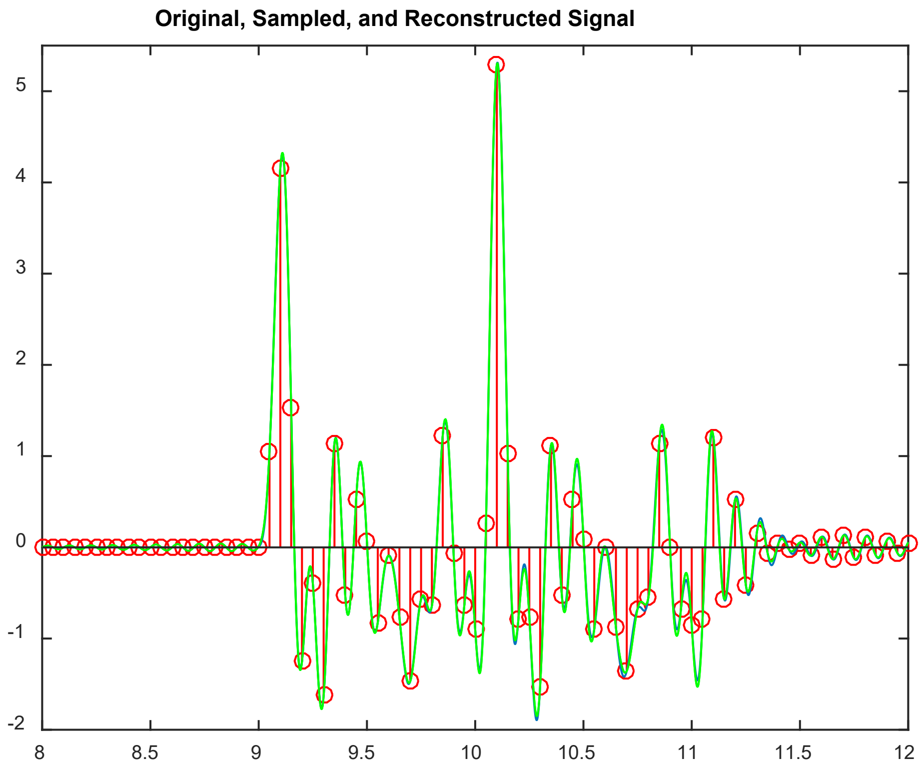
# Sampling for Discrete Signals



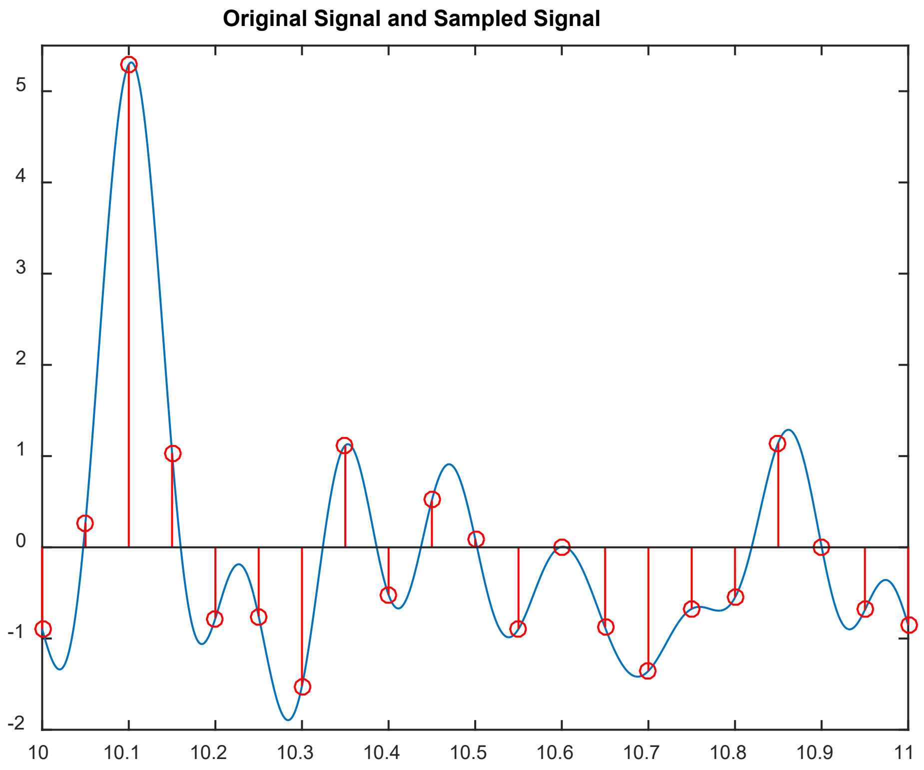

Sampled at 20 Hz

Signal constructed from 10 freq. components (1 to 10 Hz )

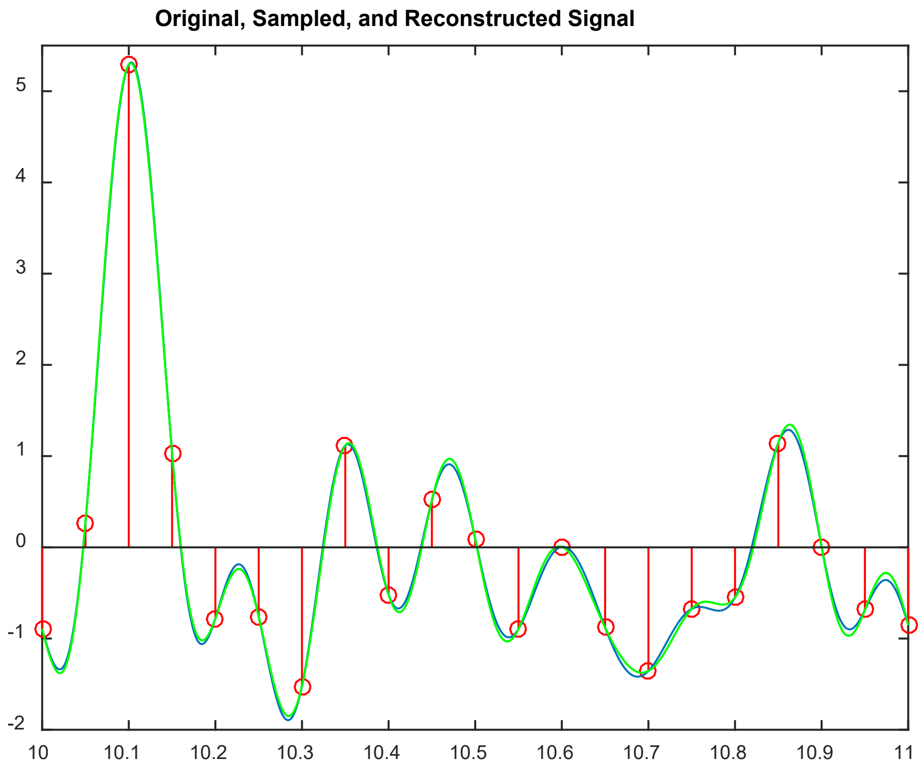
# Sampling for Discrete Signals



# Sampling for Discrete Signals



# Sampling for Discrete Signals



# Simple Example of Aliasing

- Consider the following signal:

$$x(t) = 10\sin(2\pi f_1 t) + 8\sin(2\pi f_2 t) + 4\sin(2\pi f_3 t)$$

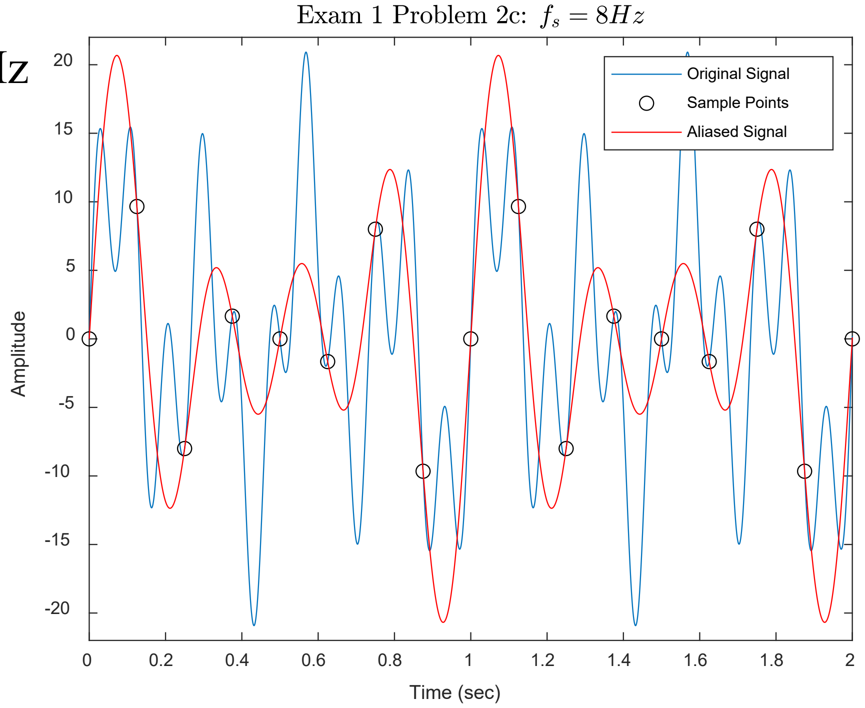
For $f_1 = 4$ Hz; $f_2 = 11$ Hz; $f_1 = 2$ Hz;

$$x(t) = 10\sin(2\pi 4t) + 8\sin(2\pi 11t) + 4\sin(2\pi 2t)$$

If you sampled at $f_s = 8$ Hz, the Nyquist frequency would be $f_{NY} = 4$ Hz
Thus, the second term would be aliased.
This can be re-written as:

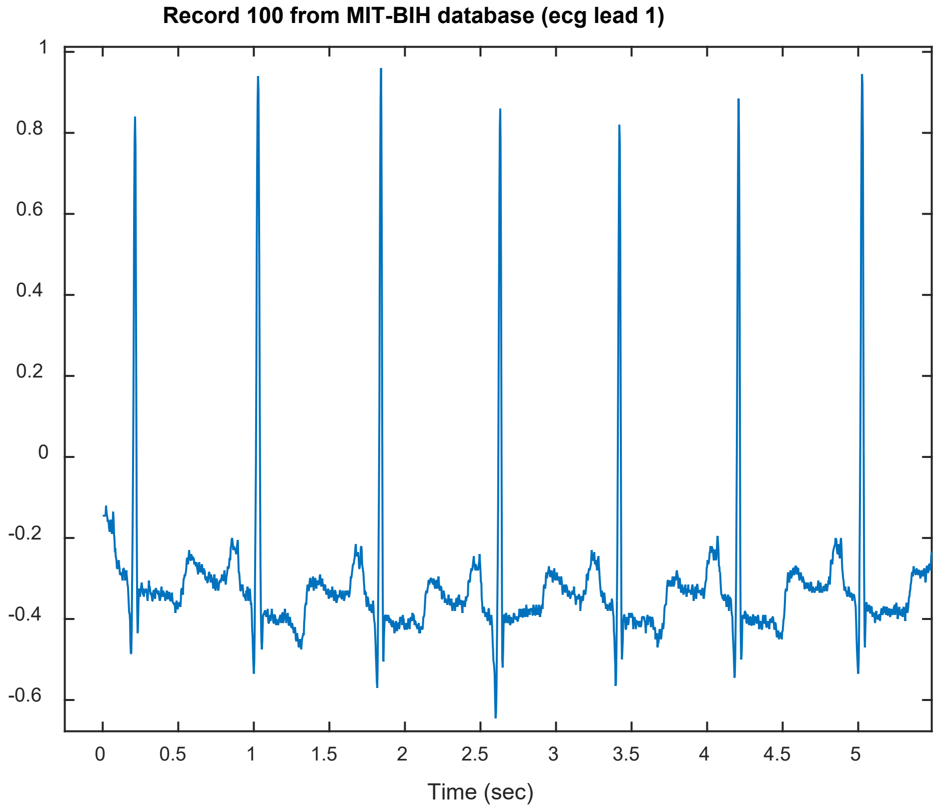
$$x(n) = 10\sin\left(2\pi\frac{4}{8}n\right) + 8\sin\left(2\pi\left(1+\frac{3}{8}\right)n\right) + 4\sin\left(2\pi\frac{2}{8}n\right)$$

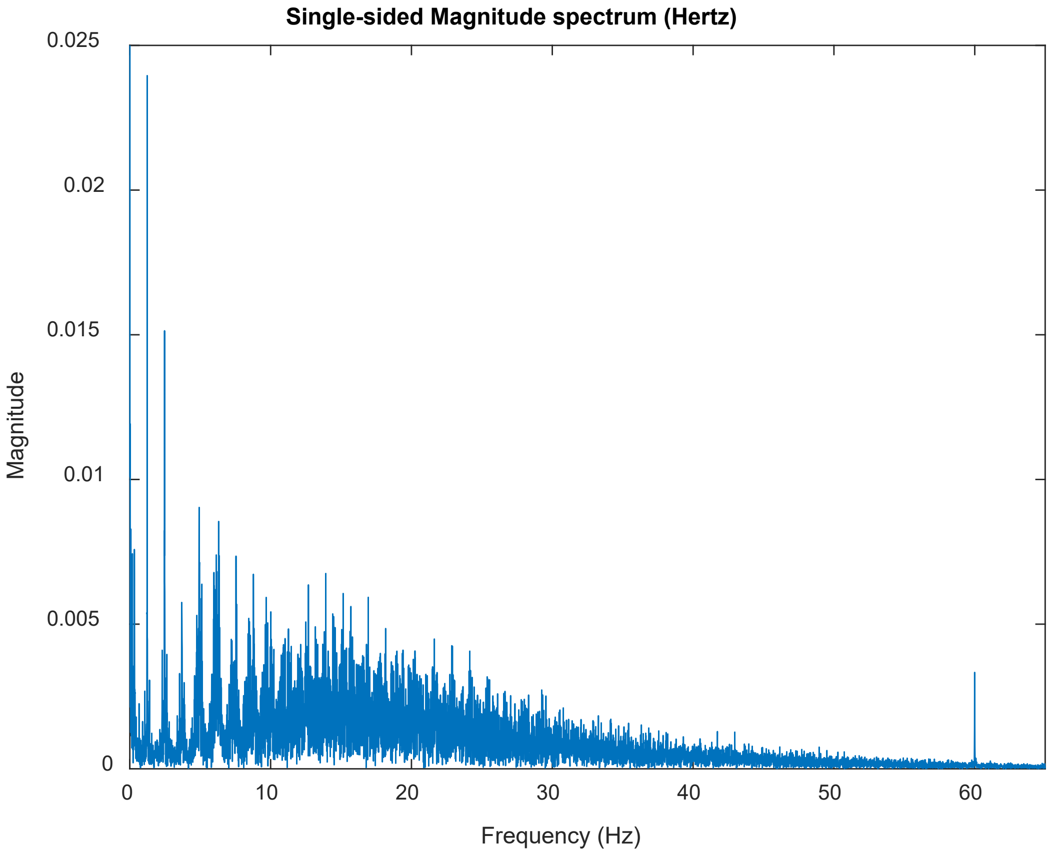
$$= 10\sin\left(2\pi\frac{4}{8}n\right) + 8\left[\sin\left(2\pi\frac{3}{8}n\right)\right] + 4\sin\left(2\pi\frac{2n}{8}\right)$$




The sampled signal looks like it could have come from:
$x(t) = 10\sin(2\pi 4t) + 8\cos(2\pi 3t) + 4\sin(2\pi 2t)$

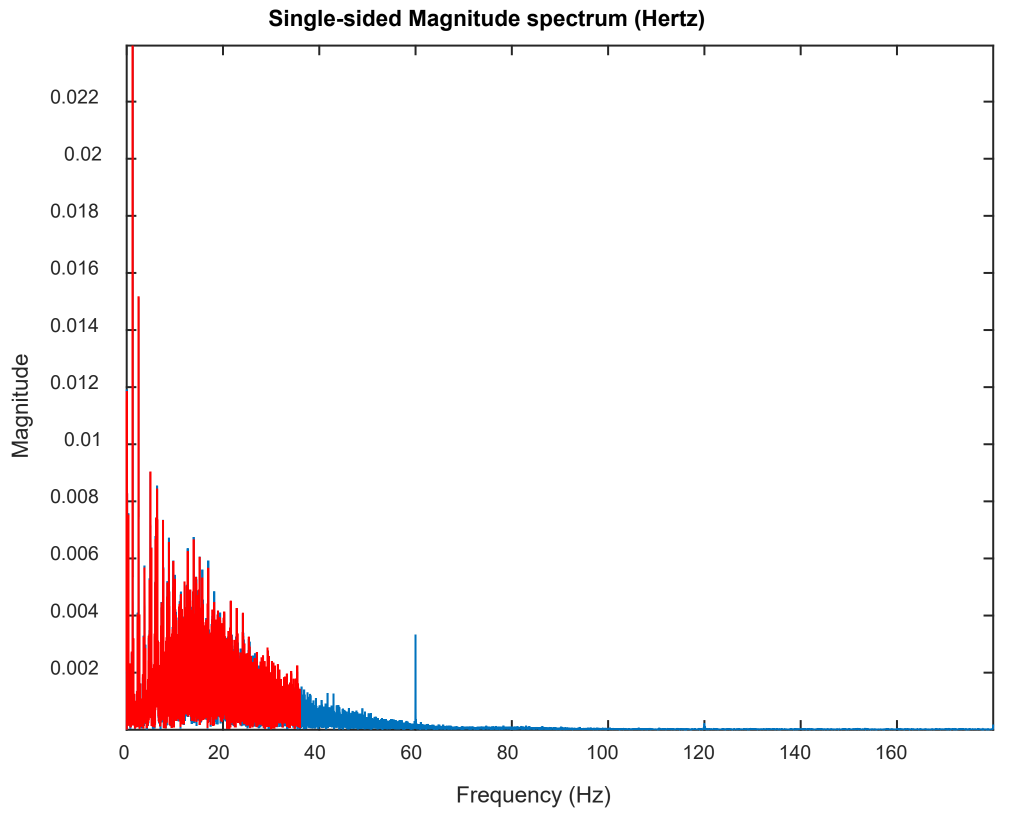
# Real signal example – ECG signal




Sampled at 360 Hz, so Nyquist frequency is 18 Hz

# Real signal example – ECG signal

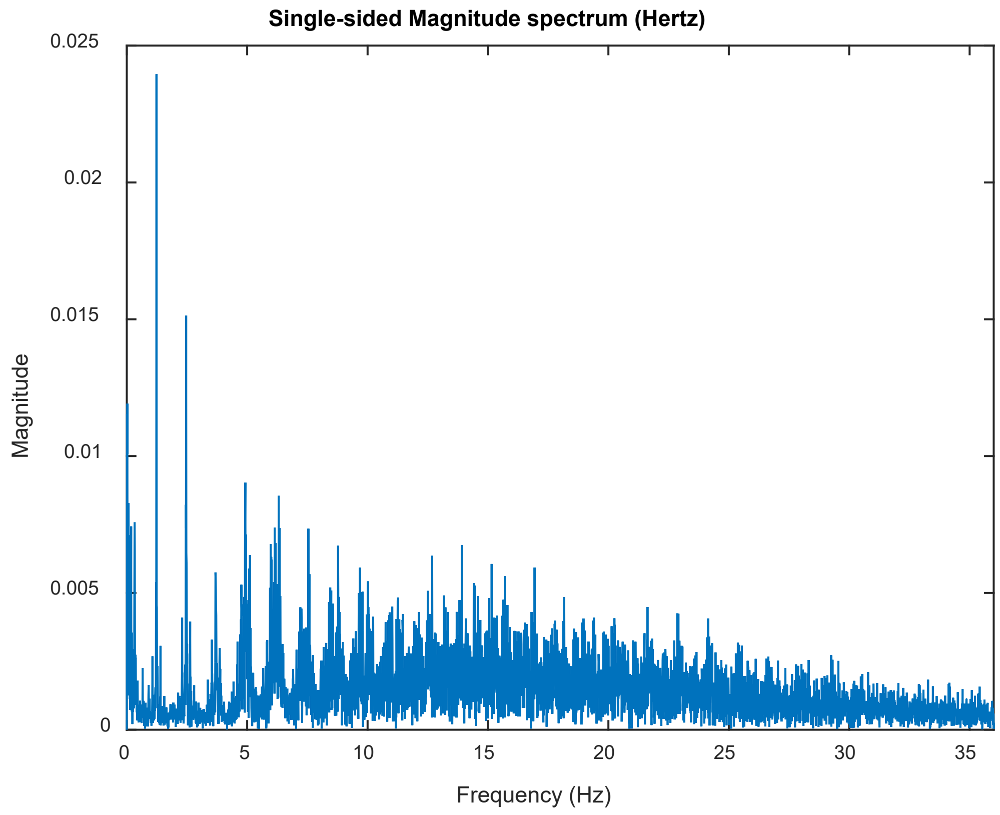



ECG signal

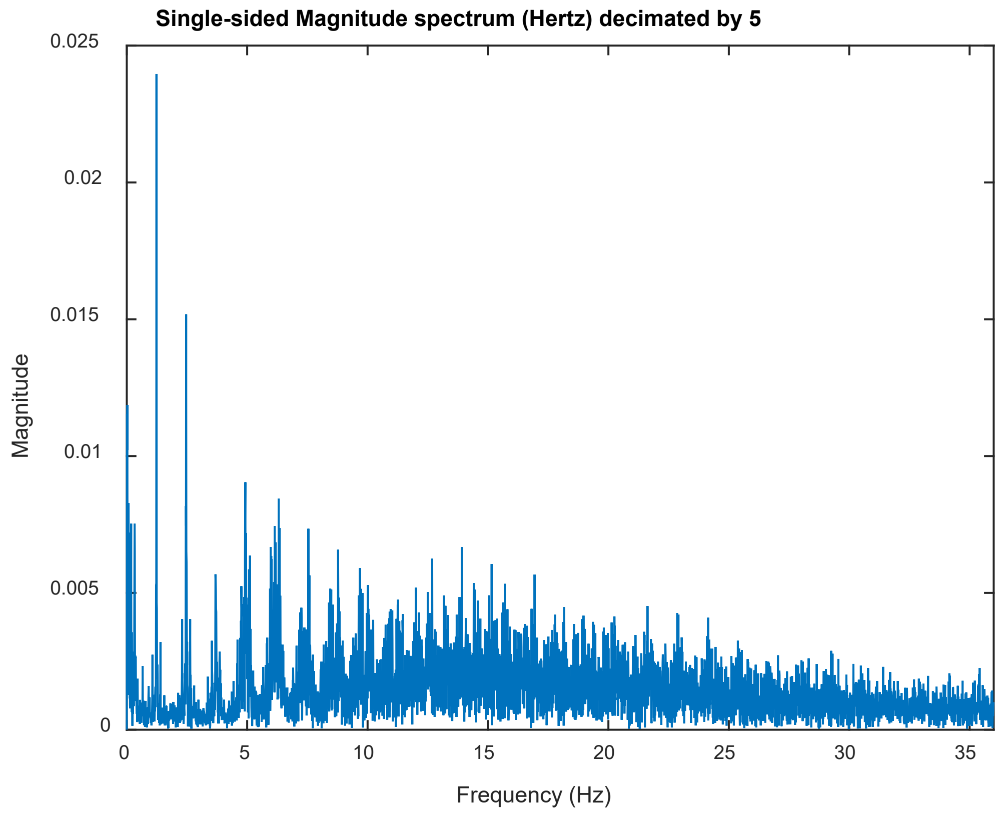
Red - Decimated by 5

Like sampling at 360/5 = 72 Hz
Max freq is 36 Hz

# Real signal example – ECG signal

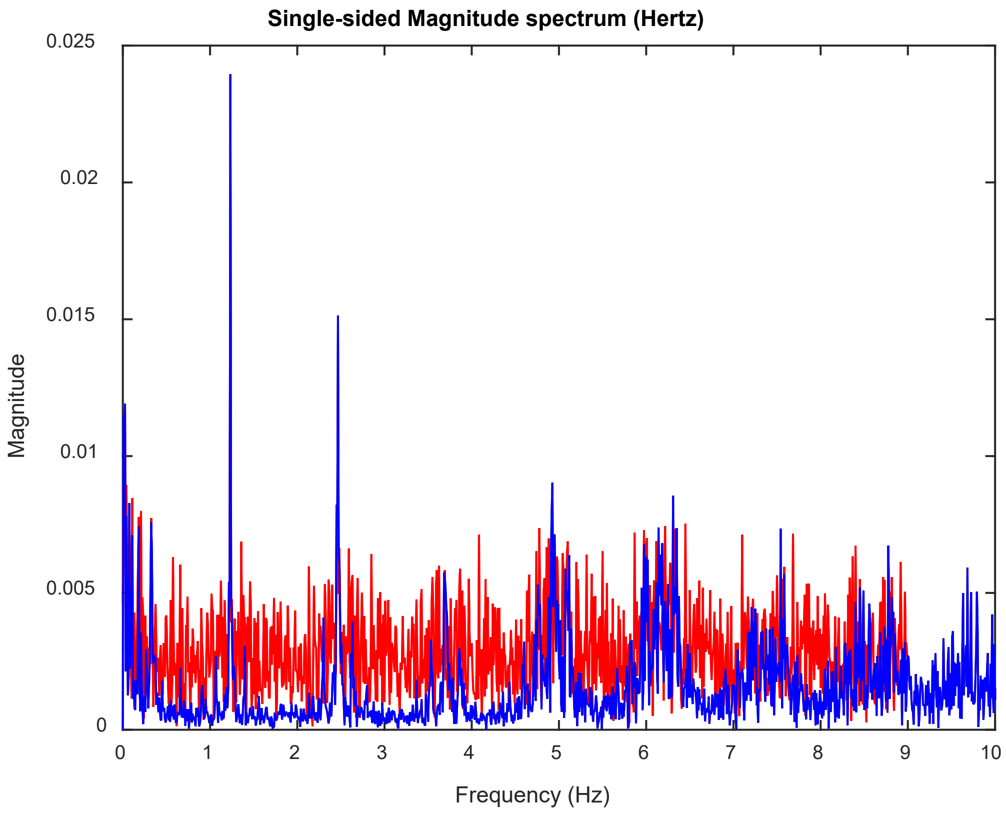


# Real signal example – ECG signal




Decimated by 5

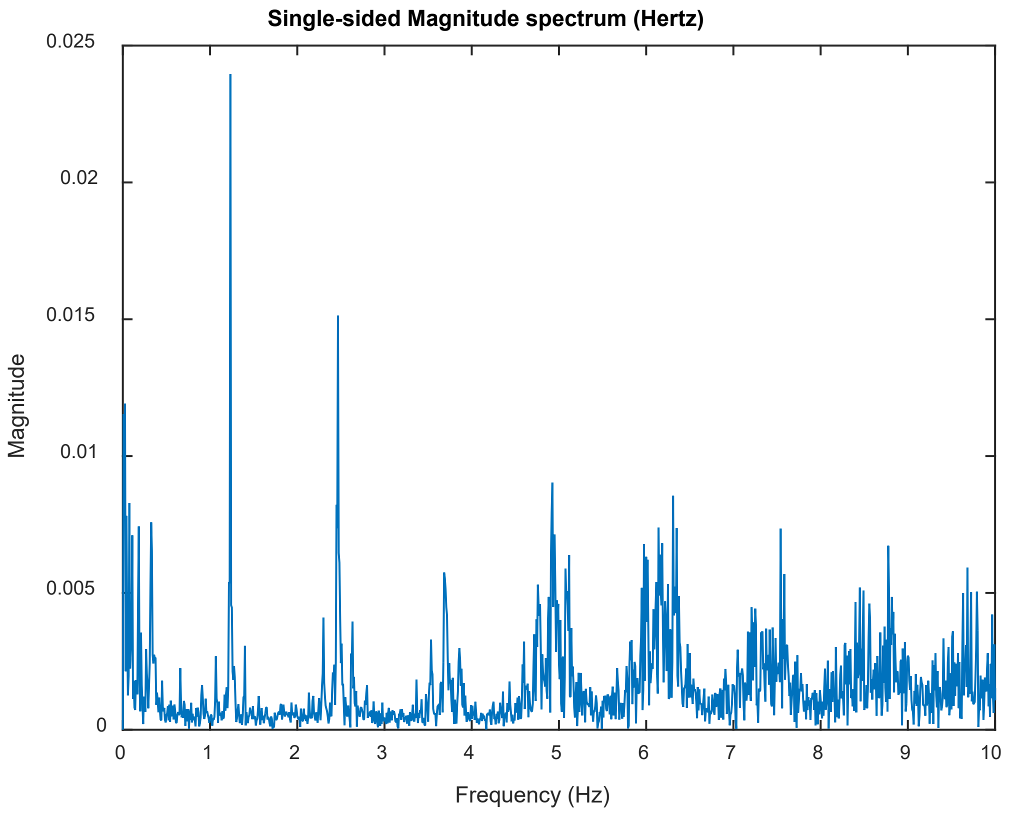
# Real signal example – ECG signal



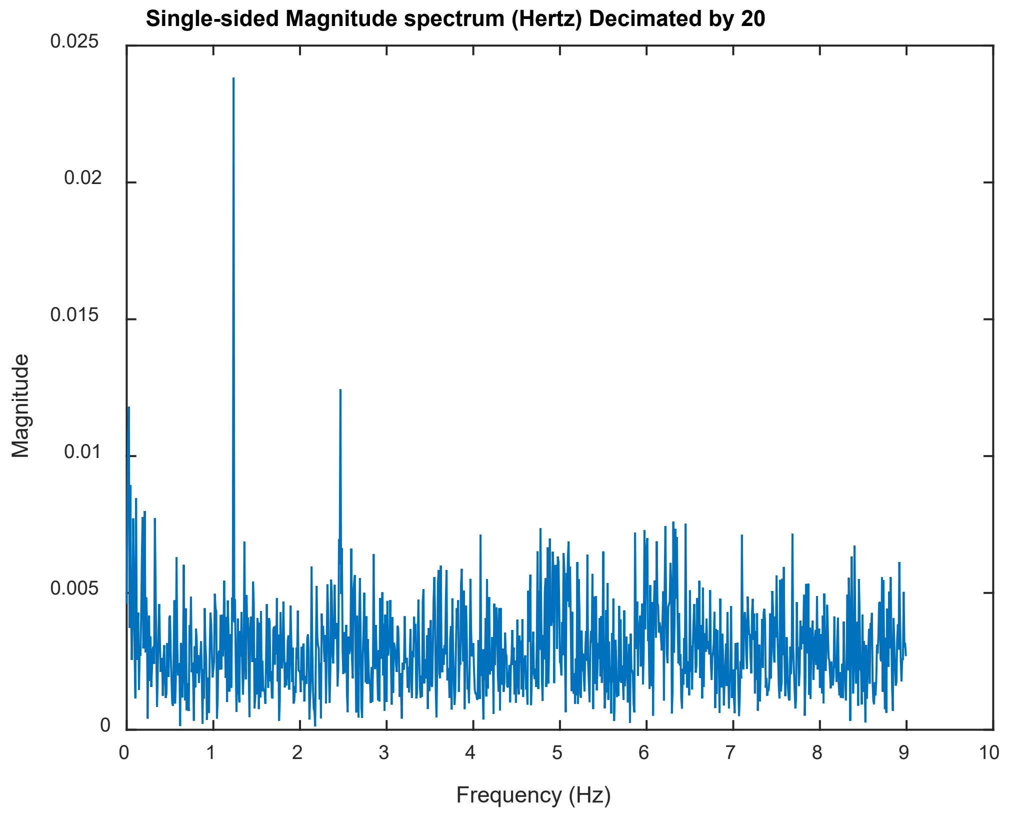

Red - Decimated by 20

Like sampling at 360/20 = 18 Hz
Max. freq is 9 Hz

# Real signal example – ECG signal

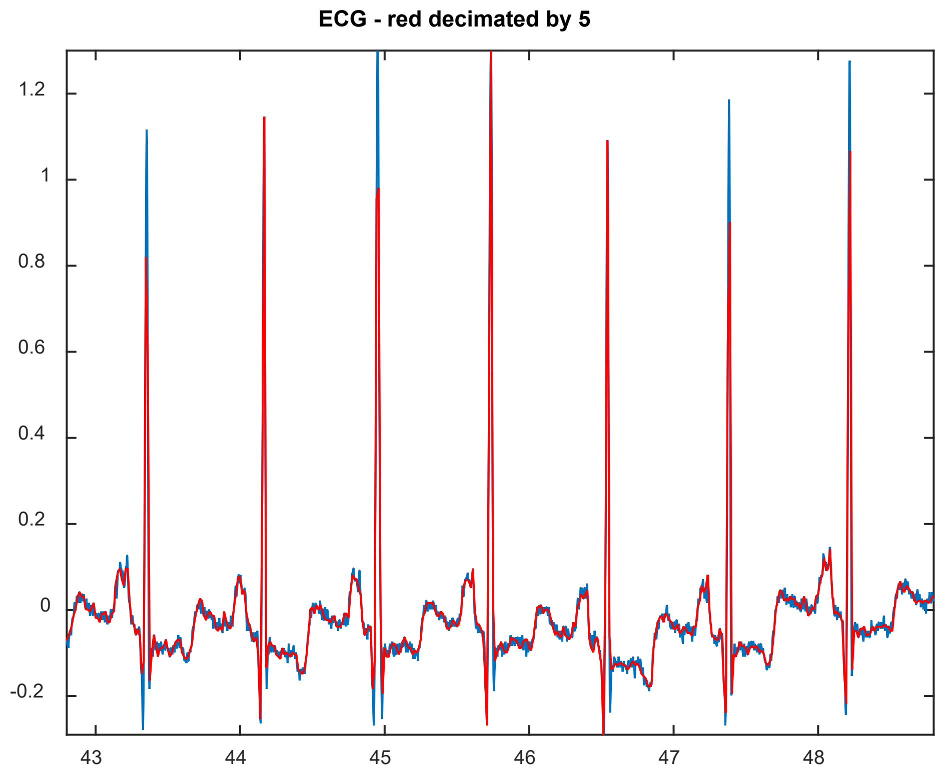


# Real signal example – ECG signal



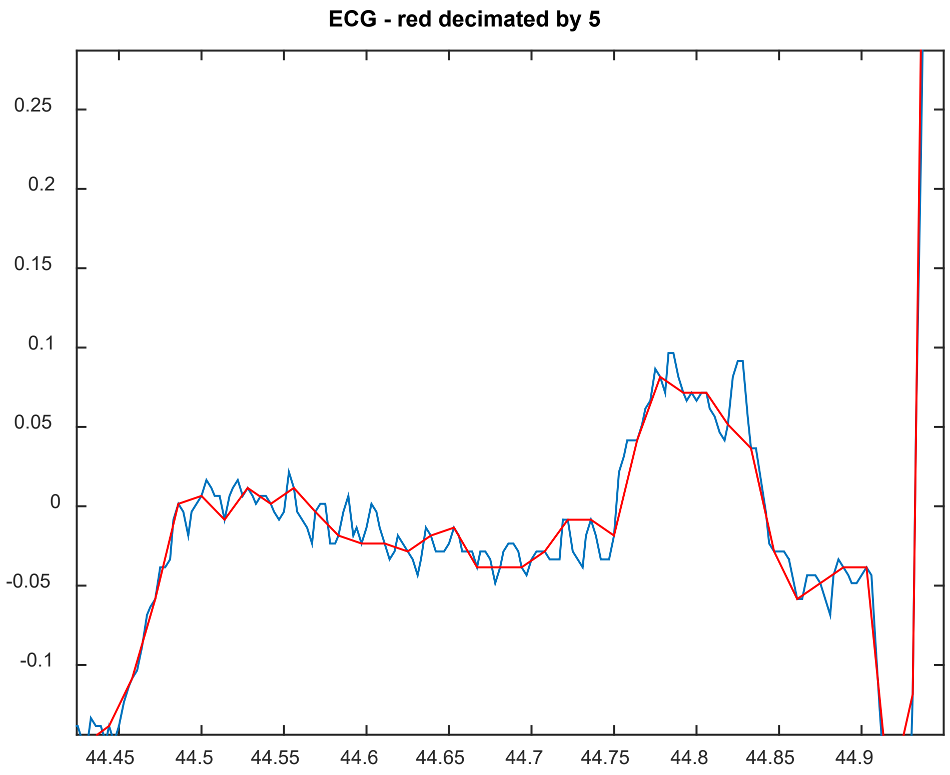

Decimated by 20

# Real signal example – ECG signal



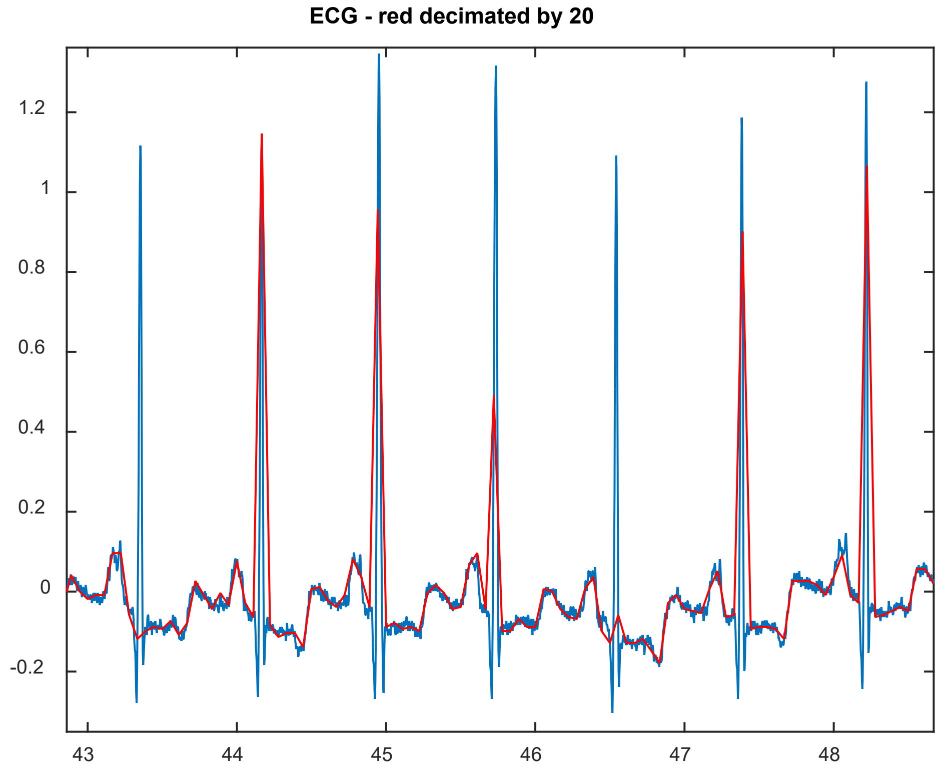

Decimated by 5

# Real signal example – ECG signal




Decimated by 5

# Real signal example – ECG signal




Decimated by 20

# What does aliasing sound like?

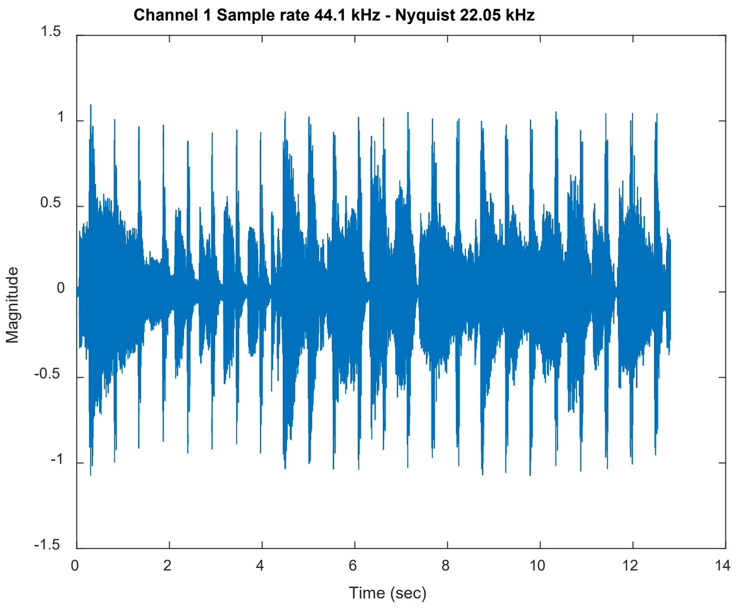
- 14 seconds of “Love Fool” by Cardigans
  - From Pandora – sampled at 44.1 kHz (CD quality)
  - Originally, there were 2 channels, since stereo – we’ll just look at one channel

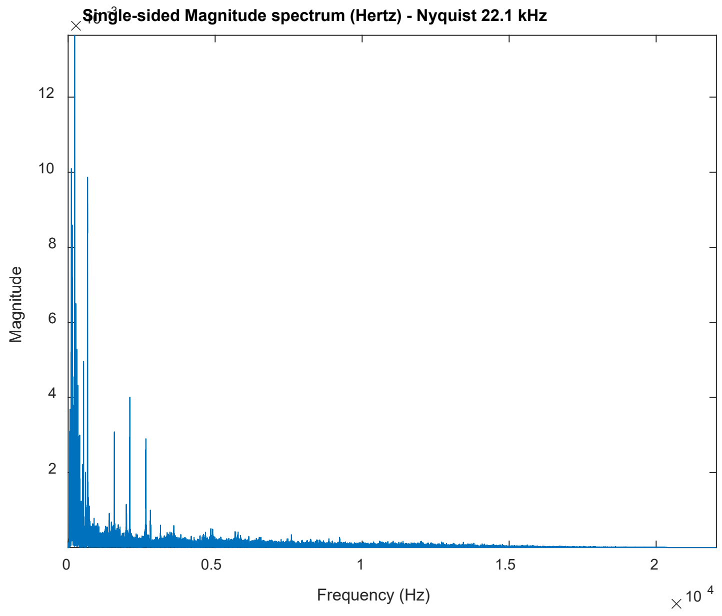
| | | |
|---|---|---|
| • Original | 44.1 kHz; | Nyquist 22.05 kHz |
| • Decimated by 5 | 8.82 kHz; | Nyquist 4.41 kHz |
| • Decimated by 10 | 4.41 kHz; | Nyquist 2.205 kHz |
| • Decimated by 15 | 2.94 kHz; | Nyquist 1.47 kHz |
| • Decimated by 30 | 1.47 kHz; | Nyquist 0.735 kHz |

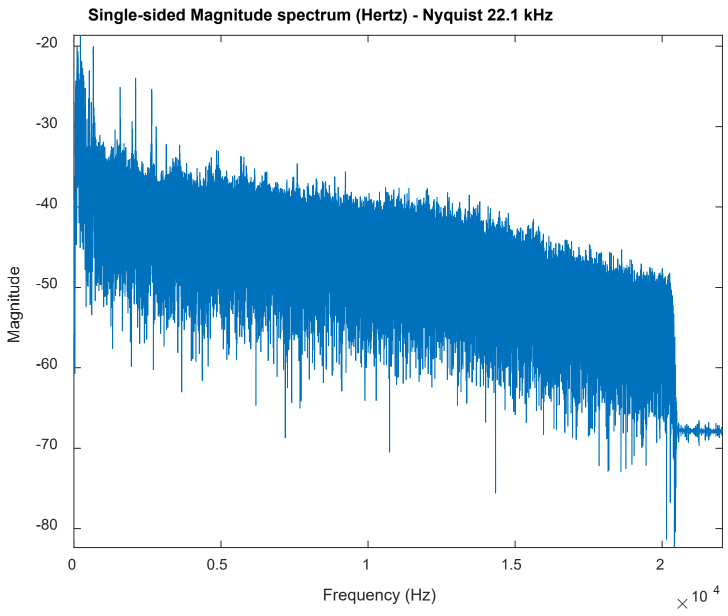
# What does aliasing sound like?

- Original: sample at 44.1 kHz; Nyquist is 22.05 kHz

    sound(ch1_part,fs)



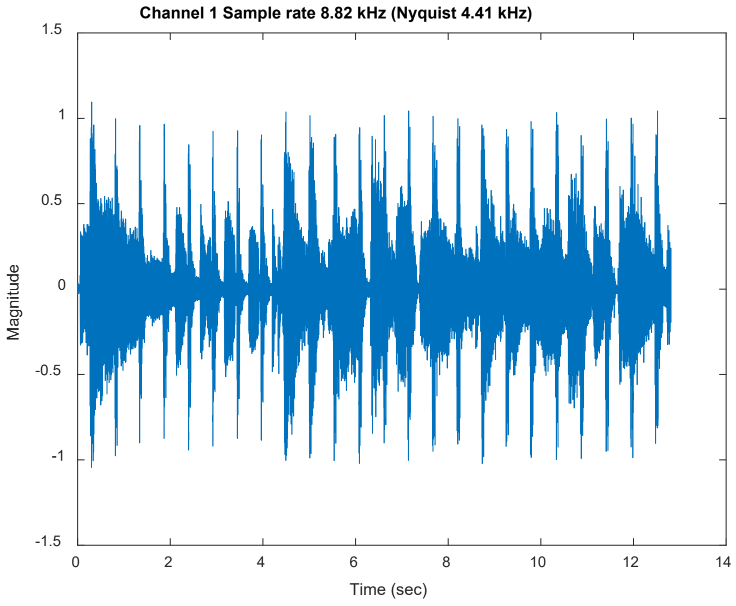



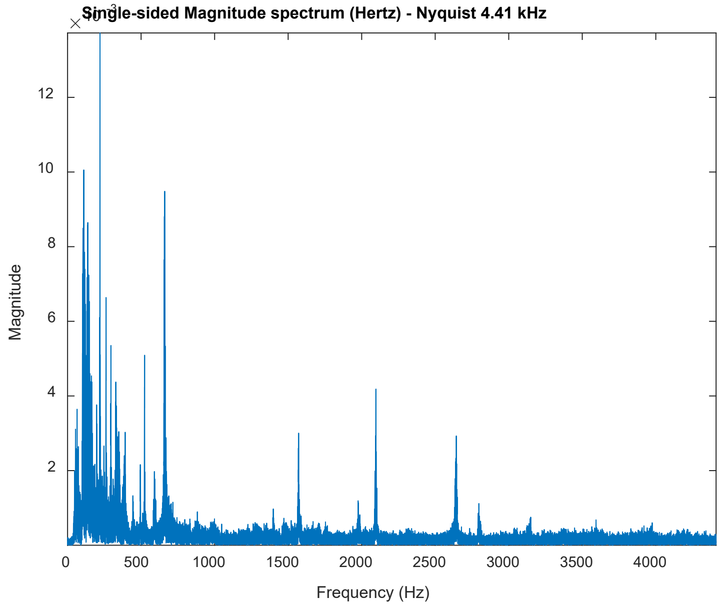



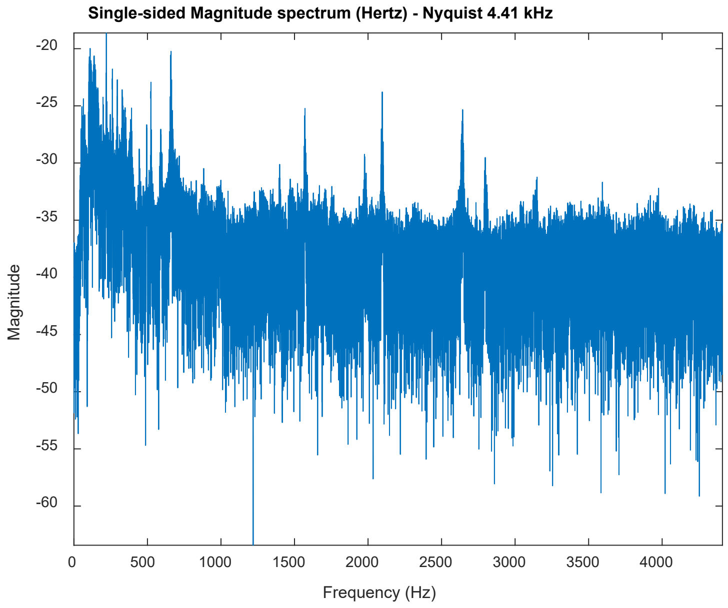
# What does aliasing sound like?

- Decimated by 5, like sampling at 8.82 kHz; Nyquist 4.41 kHz

  sound(ch1_part_sub5,fs_sub5)



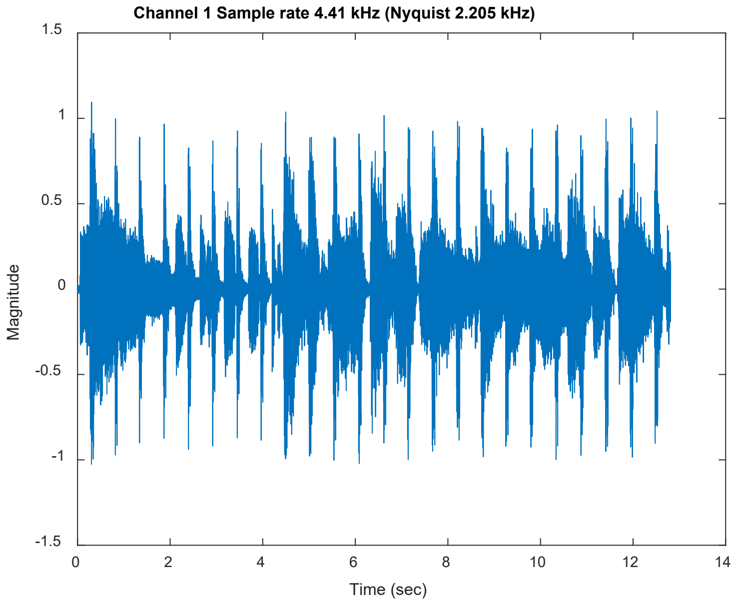



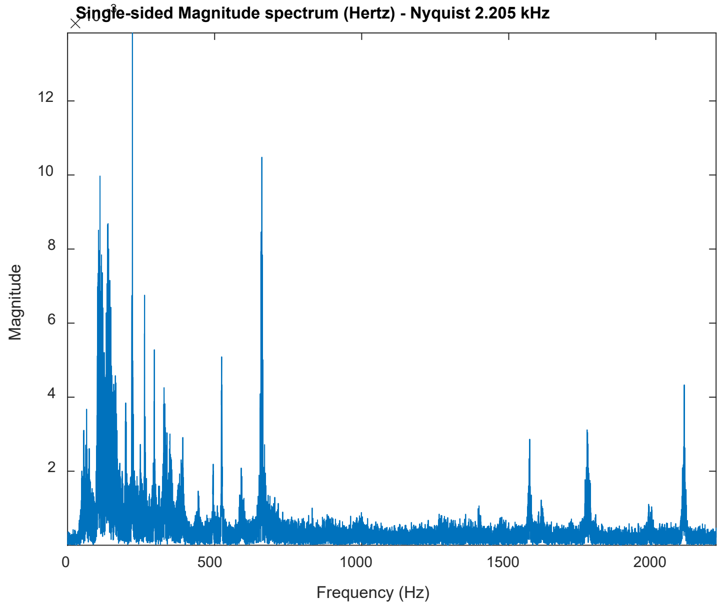



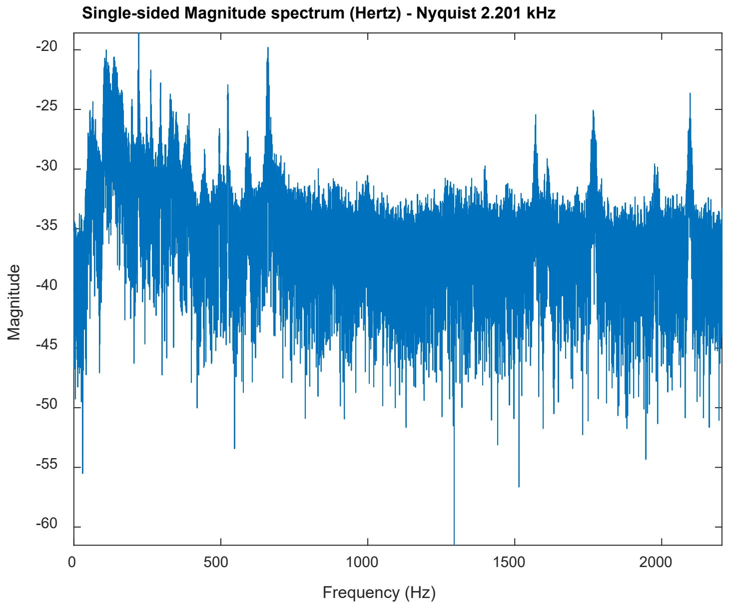
# What does aliasing sound like?

- Decimated by 10, like sampling at 4.41 kHz; Nyquist 2.205 kHz

  sound(ch1_part_sub10,fs_sub10)



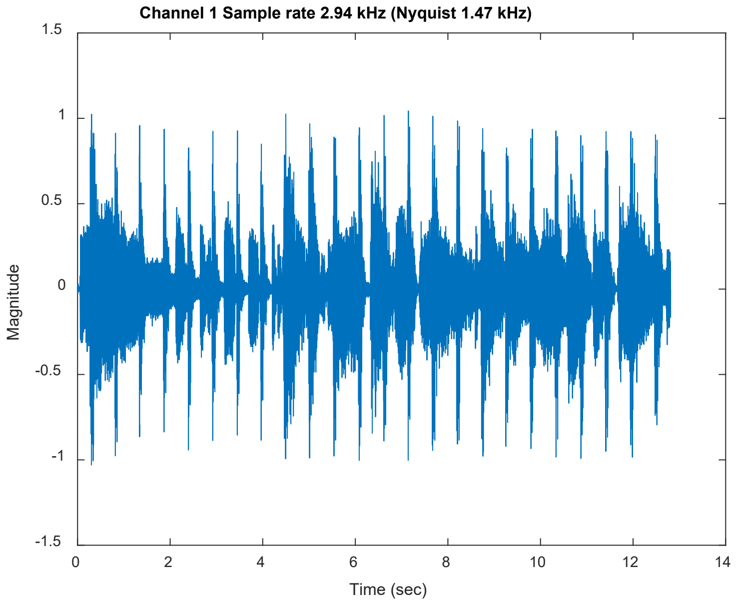



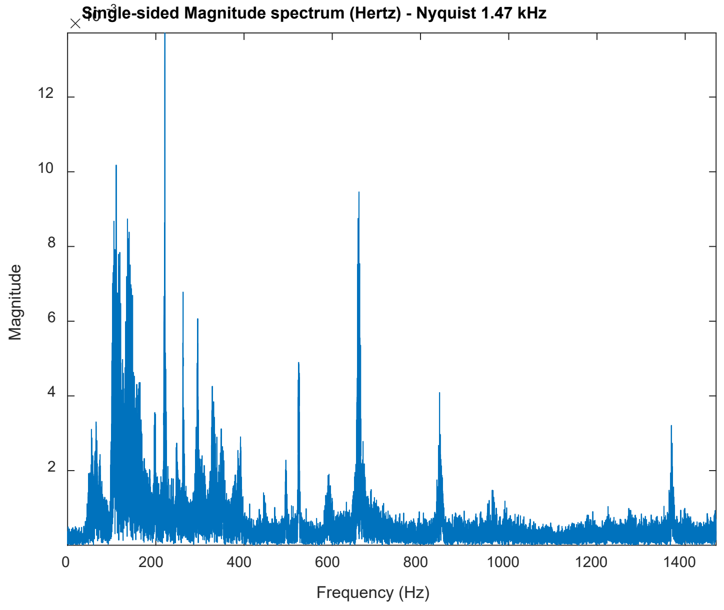



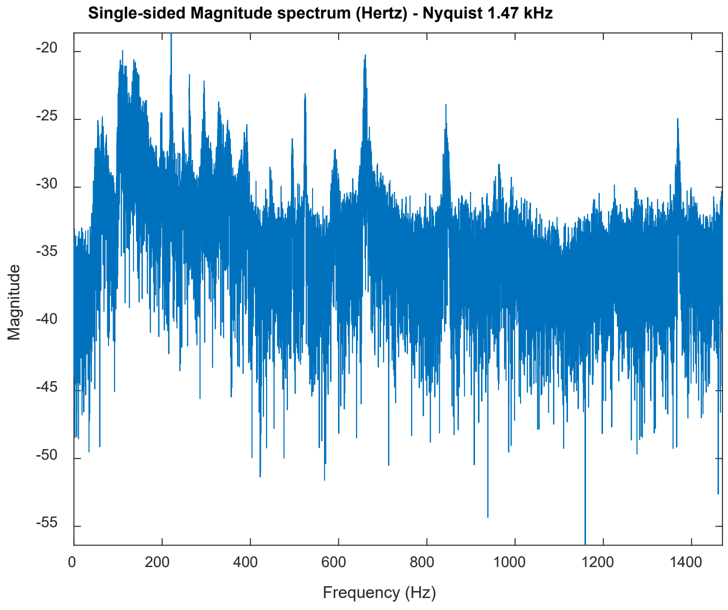
# What does aliasing sound like?

- Decimated by 15, like sampling at 2.94 kHz; Nyquist 1.47 kHz

  sound(ch1_part_sub15,fs_sub15)



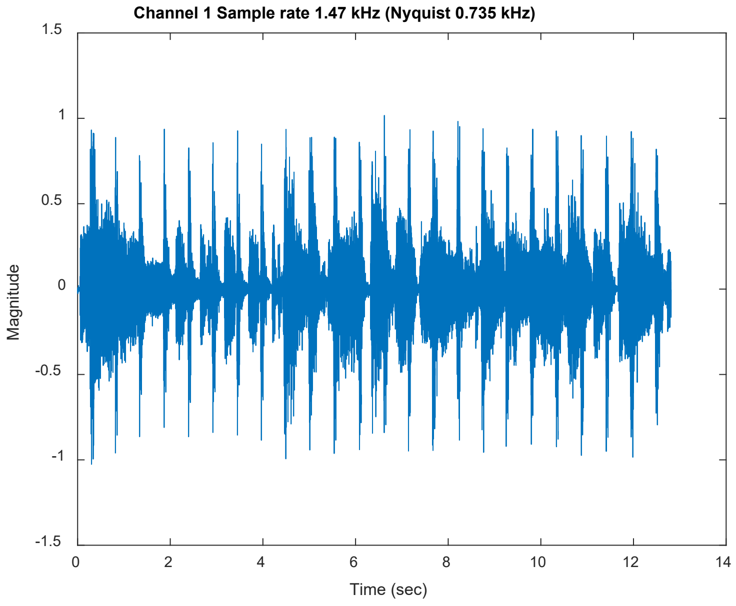



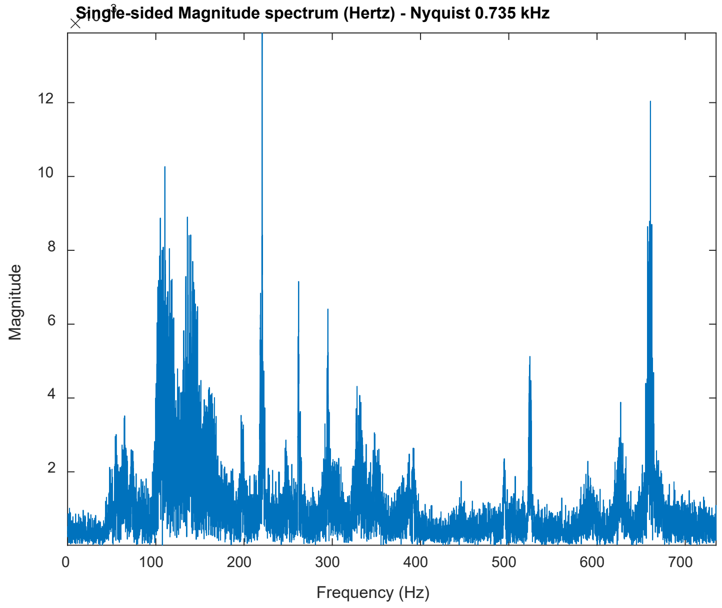



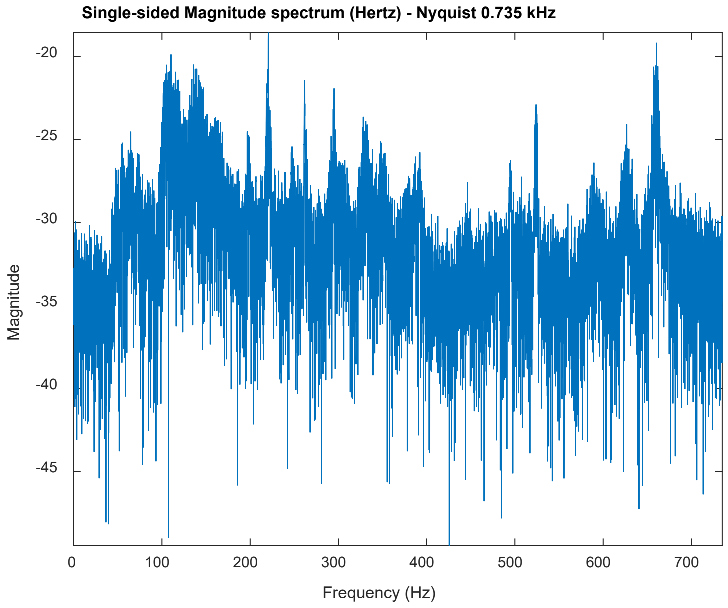
# What does aliasing sound like?

- Decimated by 30, like sampling at 1.47 kHz; Nyquist 0.735 kHz

  sound(ch1_part_sub30,fs_sub30)



# Filtering

- What to do when signals are not band-limited?
  - Apply a low-pass filter to eliminate frequencies above the range of interest.
    - Anti-aliasing filter
  - Then sample with sampling rate that matches the band-width of the filter.
- Example
  - Run filterDesigner
  - Create FIR filter for sub15 (low pass filter with cutoff at 1500 Hz)

Demonstrate:

ch1_part_sub15 is decimated by 15 (Nyquist 1.47 kHz)

Load ch1_part_filt_1500_sub15.mat
ch1_part_filt_1500_sub15 is decimated by 15 on low-pass filtered signal