Lecture 5: Sampling & Aliasing
Practice Exercises

Konstantinos Chatzilygeroudis
costashatz@upatras.gr

21 April 2026

1 Coding Exercises

Question 1.1

Consider the continuous-time signal

$$x(t) = \cos(2\pi \cdot 3\, t).$$

Write Python code using `numpy` and `matplotlib` to:

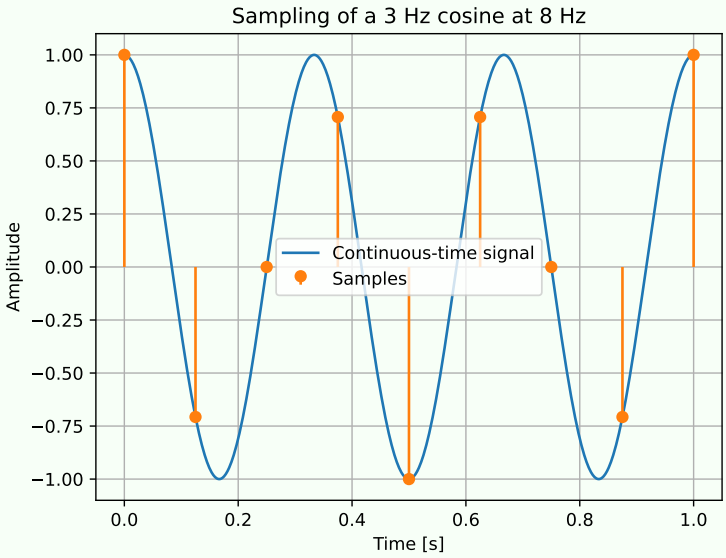
1. Plot the continuous-time signal over the interval $0 \leq t \leq 1$.
2. Sample it with sampling frequency $f_s = 8$ Hz.
3. Plot the samples on top of the continuous-time signal.
4. Determine from the plot whether aliasing occurs.

Solution (Code)

```python
import numpy as np
import matplotlib.pyplot as plt

# Continuous-time signal
f0 = 3  # Hz
t = np.linspace(0, 1, 1000)
x = np.cos(2 * np.pi * f0 * t)

# Sampling
fs = 8  # Hz
Ts = 1 / fs
n = np.arange(0, int(1 / Ts) + 1)
tn = n * Ts
xn = np.cos(2 * np.pi * f0 * tn)

# Plot
plt.figure()
plt.plot(t, x, label='Continuous-time signal')
plt.stem(tn, xn, linefmt='C1-', markerfmt='C1o', basefmt=' ', label='Samples')
plt.xlabel('Time [s]')
plt.ylabel('Amplitude')
plt.title('Sampling of a 3 Hz cosine at 8 Hz')
plt.grid(True)
plt.legend()
plt.show()
```

Solution

Plot:



Answers:
The signal frequency is $f_0 = 3$ Hz and the sampling frequency is $f_s = 8$ Hz.
The Nyquist frequency is

$$f_N = \frac{f_s}{2} = 4 \text{ Hz}.$$

Since

$$f_0 = 3 < 4 = f_N,$$

aliasing does not occur.
From the plot, the samples follow the oscillations of the original cosine reasonably well, and the sampled sequence represents the correct 3 Hz sinusoid.
Thus, sampling at 8 Hz is sufficient for this signal.

Question 1.2

Consider the continuous-time signal

$$x(t) = \cos(2\pi \cdot 7\, t).$$

Write Python code using `numpy` and `matplotlib` to:

1. Plot the continuous-time signal over the interval $0 \leq t \leq 1$.
2. Sample it with sampling frequency $f_s = 8$ Hz.
3. Plot the samples together with the original signal.
4. Show that the samples are consistent with a lower-frequency aliased signal.

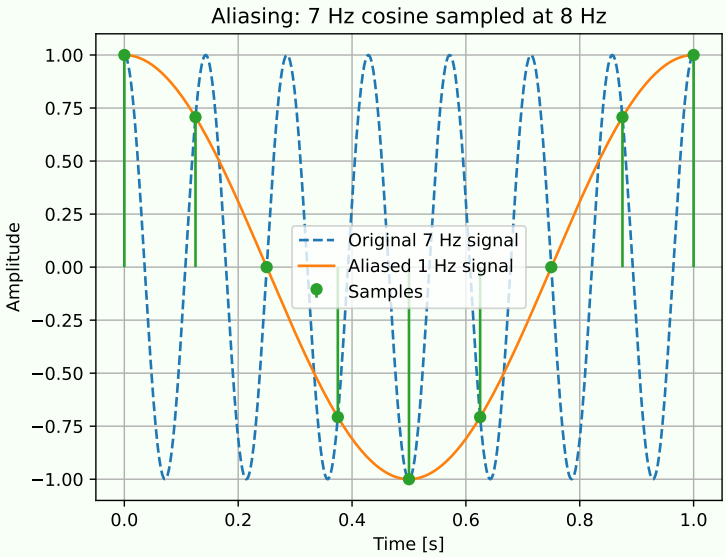
Solution (Code)

```
import numpy as np
import matplotlib.pyplot as plt

# Original signal
f0 = 7  # Hz
t = np.linspace(0, 1, 2000)
x = np.cos(2 * np.pi * f0 * t)

```

```python
# Sampling
fs = 8  # Hz
Ts = 1 / fs
n = np.arange(0, int(1 / Ts) + 1)
tn = n * Ts
xn = np.cos(2 * np.pi * f0 * tn)

# Aliased signal
f_alias = abs(f0 - fs)
x_alias = np.cos(2 * np.pi * f_alias * t)

# Plot
plt.figure()
plt.plot(t, x, '--', label='Original 7 Hz signal')
plt.plot(t, x_alias, label='Aliased 1 Hz signal')
plt.stem(tn, xn, linefmt='C2-', markerfmt='C2o', basefmt=' ', label='Samples')
plt.xlabel('Time [s]')
plt.ylabel('Amplitude')
plt.title('Aliasing: 7 Hz cosine sampled at 8 Hz')
plt.grid(True)
plt.legend()
plt.show()
```

Solution

Plot:



Answers:

Here,

$$f_0 = 7 \text{ Hz}, \qquad f_s = 8 \text{ Hz}.$$

The Nyquist frequency is

$$f_N = \frac{8}{2} = 4 \text{ Hz}.$$

Since

$$7 > 4,$$

aliasing occurs.

The aliased frequency is

$$f_{\text{alias}} = |7 - 8| = 1 \text{ Hz}.$$

The plot shows that the sample points taken from the 7 Hz signal also lie on a 1 Hz cosine. Therefore, after sampling, the 7 Hz signal is indistinguishable from a 1 Hz signal.

This is exactly the aliasing effect described in the notes.