Digital Signal Processing Laboratory
Department of Electrical & Computer Engineering

Digital Signal Processing

Lab 4

Sampling

Heraklion 2025
Dr. Konstantinos Karampidis

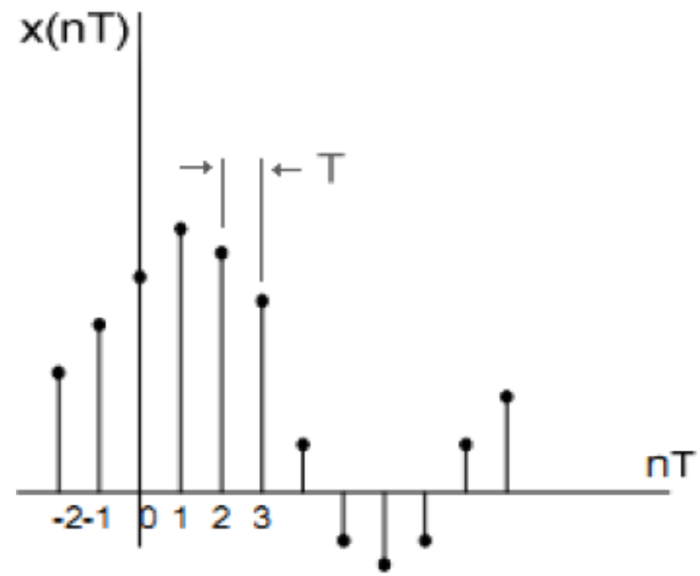
Sampling

With the process of sampling, we can convert an analogue signal into a digital signal.

During sampling, we obtain values (from the analogue signal) at regular intervals T which we call samples.

x(n) = x(nT)

where T is the sampling period



Sampling

The key question is how often should we obtain a sample?

The sampling frequency: $Fs = 1/Ts$

An analog signal x(t) with a limited range spectrum (<Fo) can be reconstructed from the samples of $x(n)=x(nTs)$ if the sampling frequency $Fs=1/Ts$ is twice the range F_0, $\mathbf{Fs \geq 2F_0}$.

In any other case there is spectrum aliasing, and the signal cannot be reconstructed.

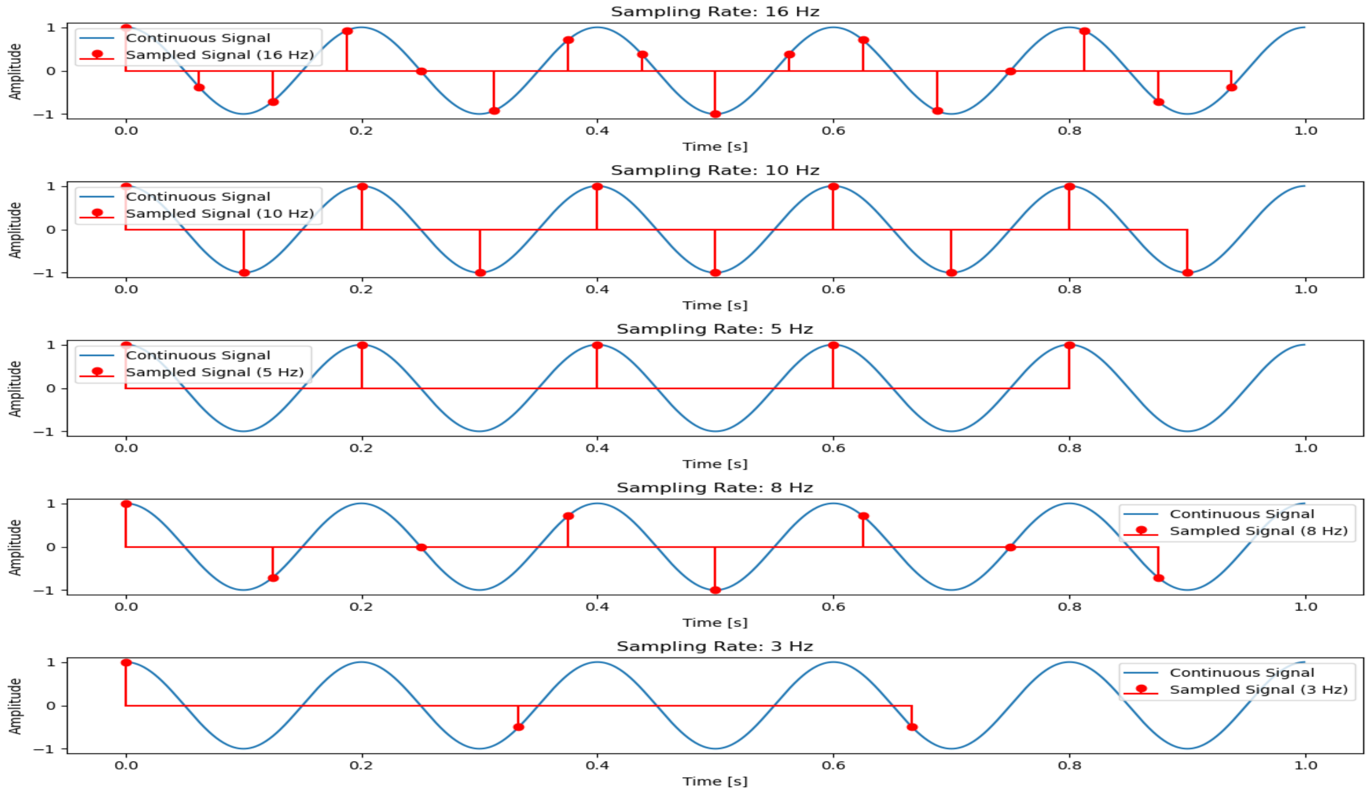
In the resulting discrete signal x(n), frequency overlaps are avoided so that the reconstruction of the original signal is possible. The minimum sampling frequency $2F_0$ is called the Nyquist Rate.

Typical sampling frequencies for standard signals

Signal type	Maximum Signal Frequency	Sampling Frequency
Geophysical	500Hz	1KHz
Biomedical	1KHz	2KHz
Engineering	2KHz	4KHz
Speech	4KHz	8KHz
Audio	20KHz	40KHz
Video	4MHz	8MHz

Example 1

Let's assume a signal with a maximum frequency of 5 Hz. The figure below shows the representation of the signal for different sampling frequency.

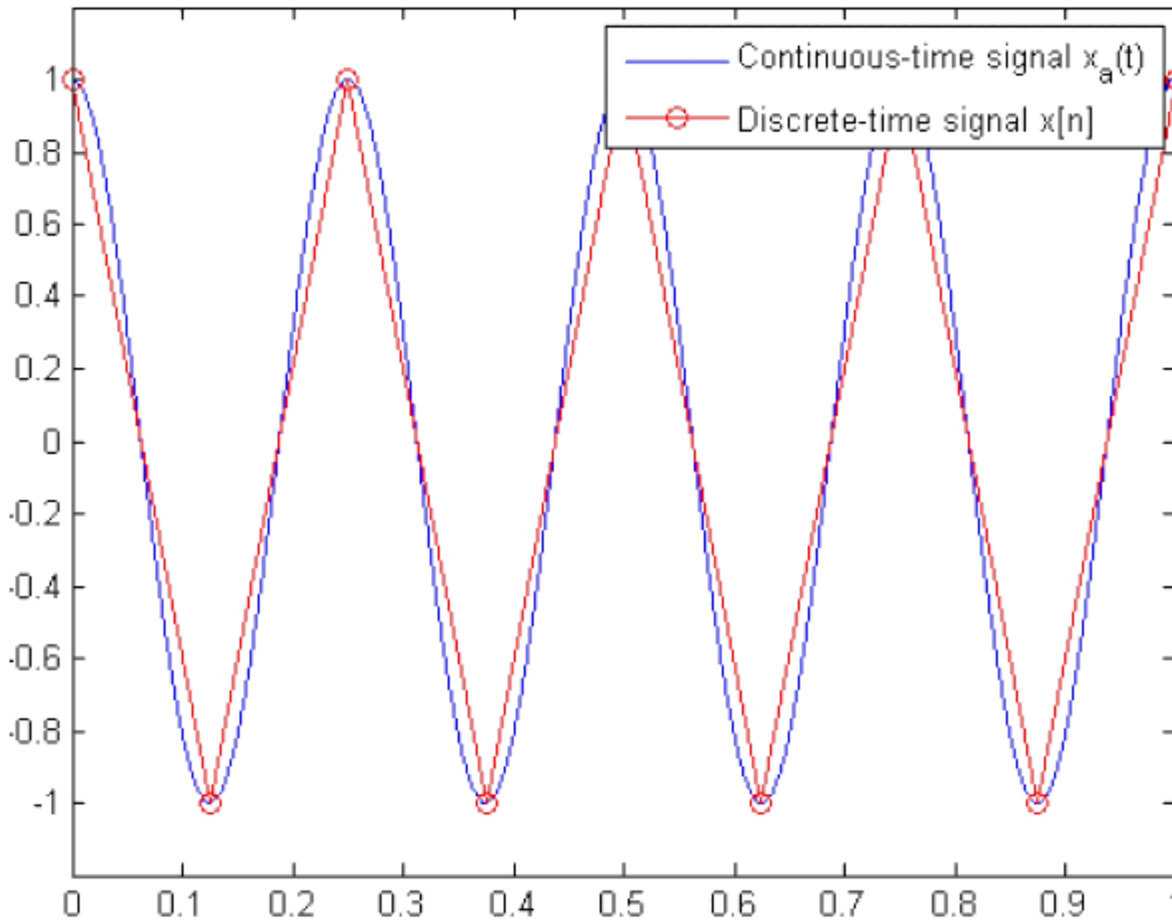


Example 2

Analog signal:

$X_a(t) = \cos(2\pi ft)$;
where f=4Hz,
t=[0,1]

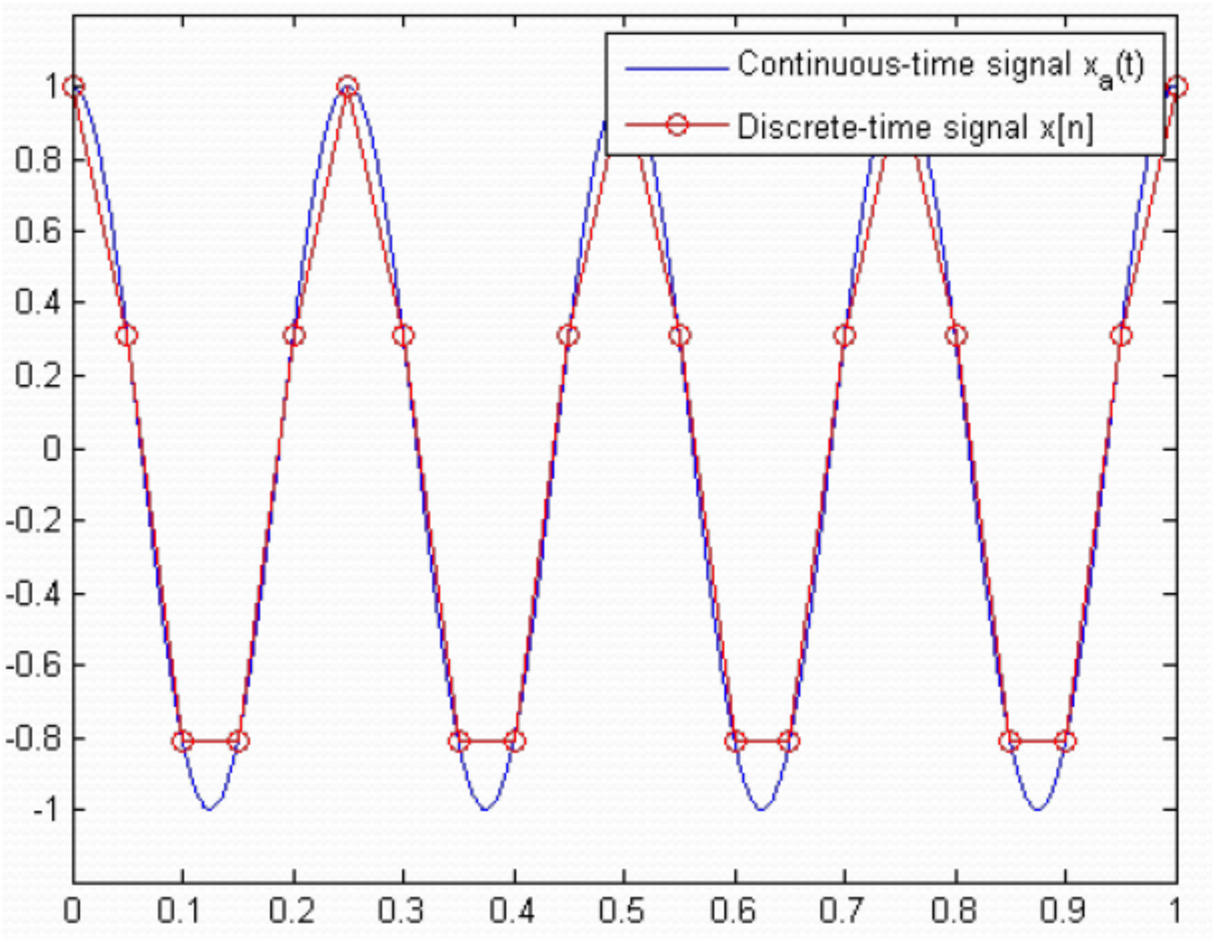
Sampling:
Ts = 0.125
Fs = 1/Ts = 8Hz



Example 3

$X_a(t) = \cos(2\pi ft)$;
where f=4Hz, t=[0,1]

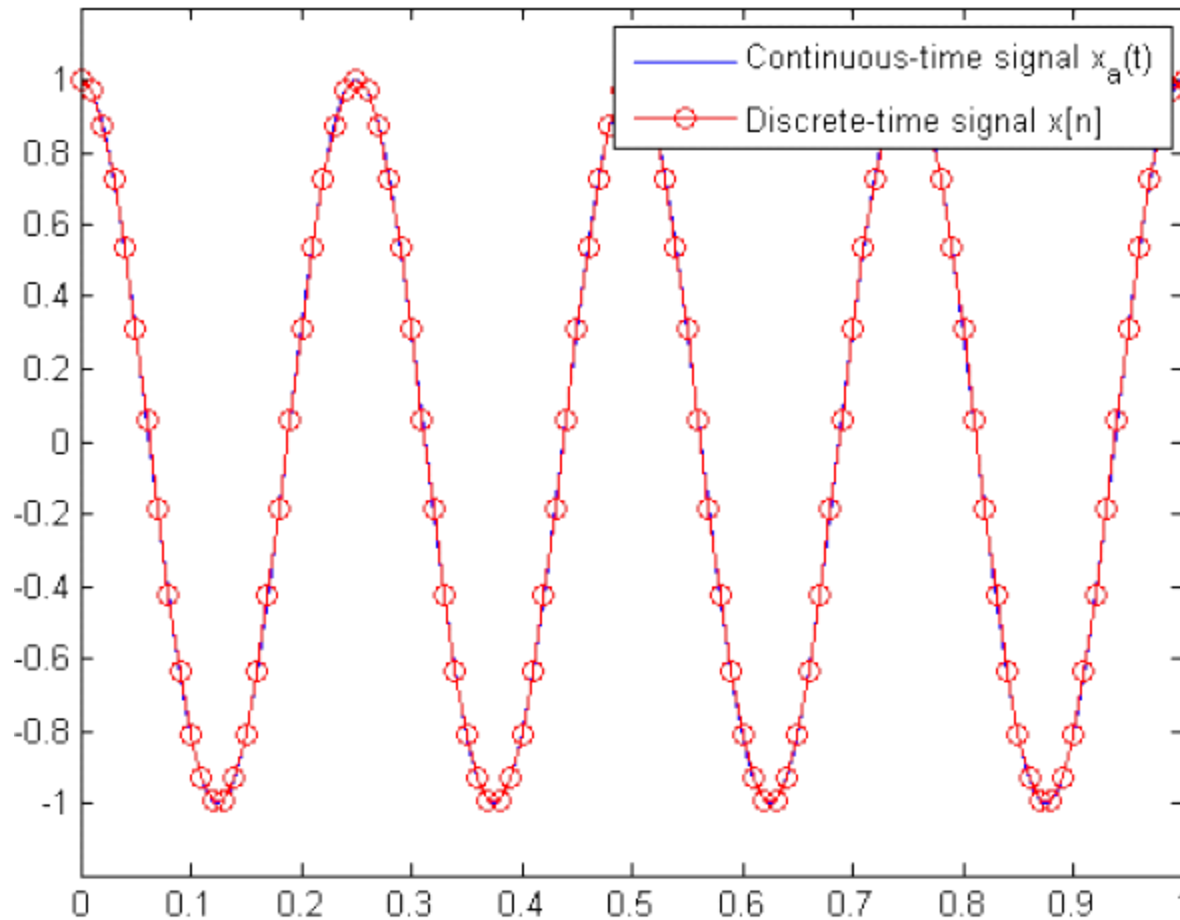
Sampling:
Ts = 0.05
Fs = 1/Ts = 20Hz



Example 4

$X_a(t) = \cos(2\pi ft);$
where f=4Hz,
t=[0,1]

Sampling:
Ts = 0.01
Fs = 1/Ts = 100Hz

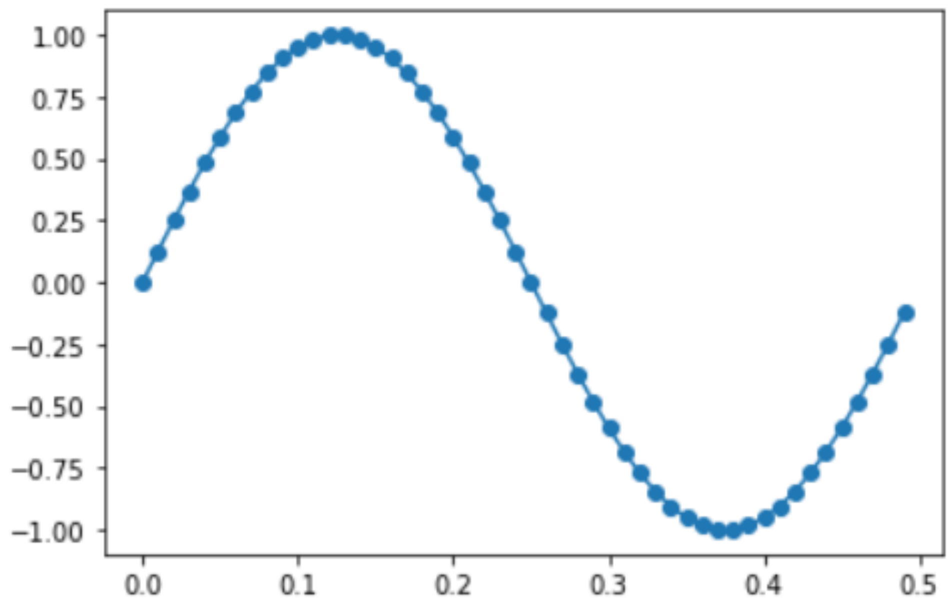


Exercise 1

Construct and plot a sine sequence of 50 samples, with a frequency f=2Hz and a sampling period T=0.01 sec (fs=100 samples/sec).

Solution

```
f = 2
t2=np.arange(0, 0.5, 0.01)
y2 = np.sin(2 * np.pi * f * t2)

plt.plot(t2, y2, 'o-')
plt.show()
```

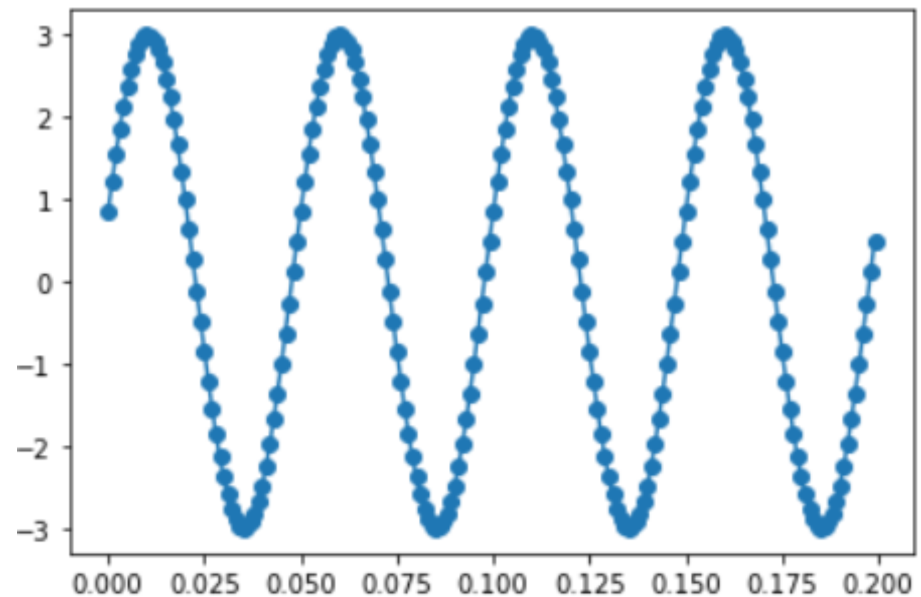


Exercise 2

Suppose we have the signal: y(t) = 3cos(40πt+5). Sample the signal at 0.5KHz and present the first 200 samples of the signal. What is the frequency of the signal?

Solution

```
t=np.arange(0, 0.2, 0.001)
y = 3*np.cos((40 * np.pi * t)+5)

plt.plot(t, y, 'o-')
plt.show()
```



Exercise 3

Suppose we have a signal:

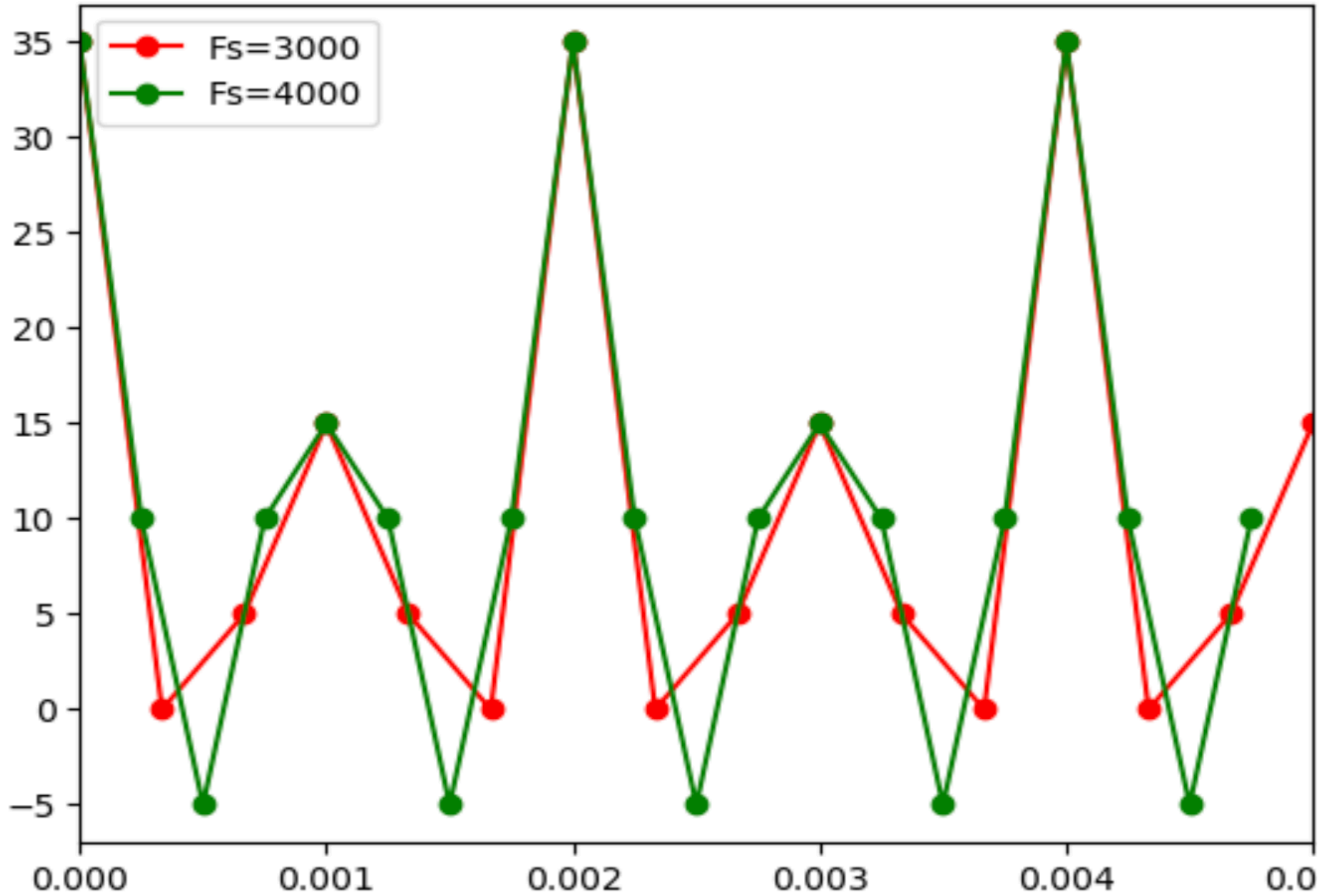
x(n) = 10 +5cos(1000πt) +15cos(2000πt) +5cos(3000πt)

Find the minimum sampling rate Nyquist Fs.

Solution

```
Fs=3000;

Ts=1/Fs;

t=np.arange(0, 20*Ts, Ts)

x=10+5*np.cos(1000*np.pi*t)+15*np.cos(2000*np.pi*t)+5*np.cos(
3000*np.pi*t);

Fs2=4000;
```

```
Ts2=1/Fs2;

t2=np.arange(0, 20*Ts2, Ts2)

x2=10+5*np.cos(1000*np.pi*t2)+15*np.cos(2000*np.pi*t2)+5*np.
cos(3000*np.pi*t2);

plt.plot(t, x, 'ro-', label="Fs=3000")

plt.plot(t2, x2, 'go-', label="Fs=4000")

plt.legend()

plt.xlim([0, 0.005])

plt.show()
```


Fs=3000
Fs=4000
35
30
25
20
15
10
5
0
−5
0.000
0.001
0.002
0.003
0.004
0.0

Exercise 4

Given a cosine signal with a maximum frequency F=1000Hz draw in the same figure:

a) the original signal

b) the resulting signal if sampled at a frequency i) less than the Nyquist rate ii) equal to the Nyquist rate iii) greater than the Nyquist rate

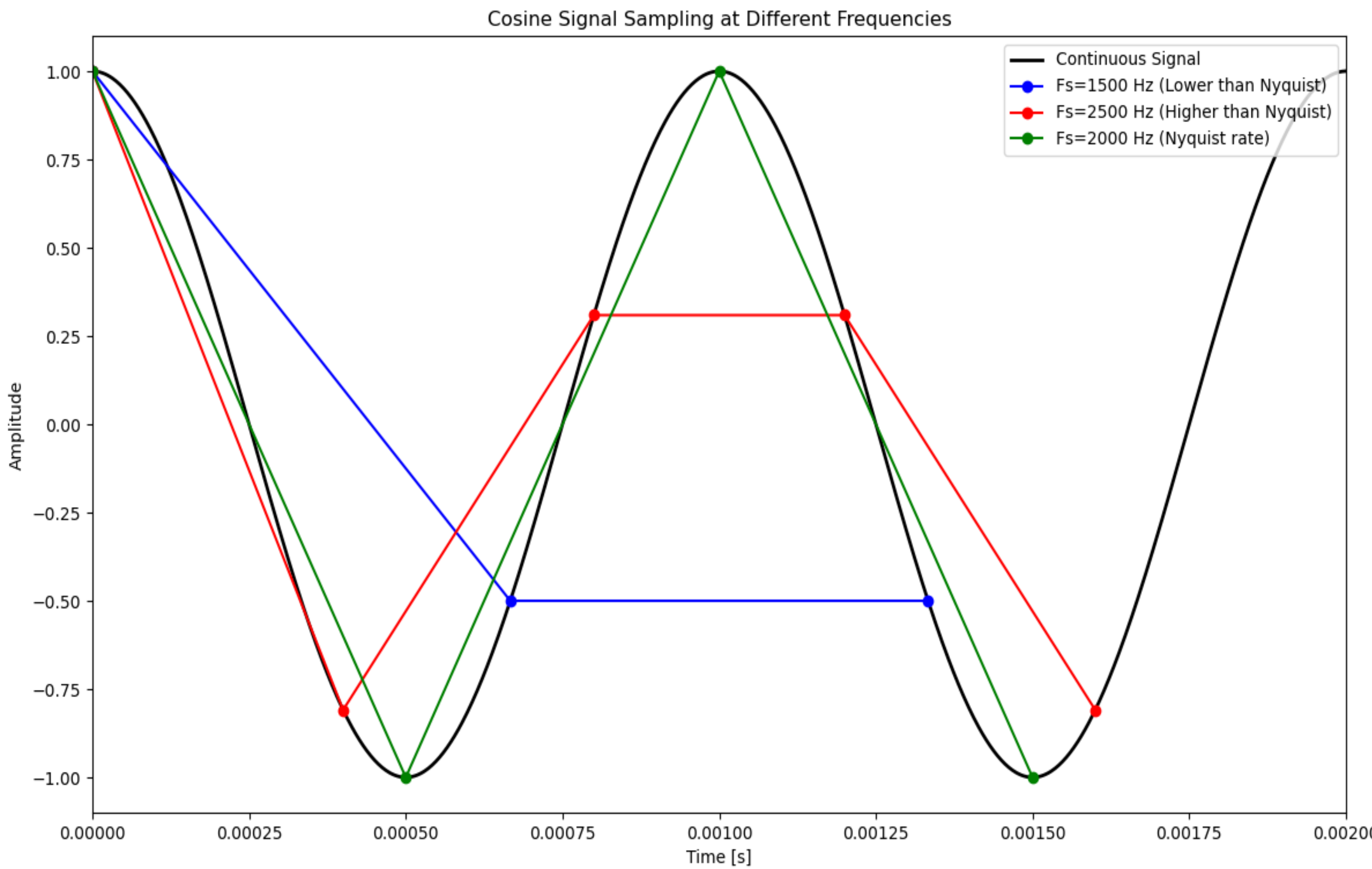
Solution

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

# Parameters
F = 1000  # Frequency of the cosine wave
T = 1 / F

# Sampling rates (lower and higher than the Nyquist rate)
Fs1 = 1500  # Lower than Nyquist rate
Fs2 = 2500  # Higher than Nyquist rate
Fs3 = 2000  # Nyquist rate
```

```
# Sampling intervals
Ts1 = 1 / Fs1
Ts2 = 1 / Fs2
Ts3 = 1 / Fs3

# Time vectors
t_continuous = np.linspace(0, 2 * T, 1000, endpoint=False)
# Adjusted time vector to focus on a smaller portion
t1 = np.arange(0, 2 * T, Ts1)
t2 = np.arange(0, 2 * T, Ts2)
t3 = np.arange(0, 2 * T, Ts3)

# Cosine signals
x_continuous = np.cos(2 * np.pi * F * t_continuous)
```

```
x1 = np.cos(2 * np.pi * F * t1)
x2 = np.cos(2 * np.pi * F * t2)
x3 = np.cos(2 * np.pi * F * t3)

# Plotting
plt.figure(figsize=(14, 8))

plt.plot(t_continuous, x_continuous, 'k-', linewidth=2, label='Continuous Signal')
plt.plot(t1, x1, 'bo-', label='Fs=1500 Hz (Lower than Nyquist)')
plt.plot(t2, x2, 'ro-', label='Fs=2500 Hz (Higher than Nyquist)')
plt.plot(t3, x3, 'go-', label='Fs=2000 Hz (Nyquist rate)')

plt.legend(loc='upper right')
```

```
plt.xlim([0, 0.002])
plt.xlabel('Time [s]')
plt.ylabel('Amplitude')
plt.title('Cosine Signal Sampling at Different Frequencies')

# Save the plot as an image
plt.savefig('cosine_signal_sampling.png')

# Show the plot
plt.show()
```


Cosine Signal Sampling at Different Frequencies
Continuous Signal
Fs=1500 Hz (Lower than Nyquist)
Fs=2500 Hz (Higher than Nyquist)
Fs=2000 Hz (Nyquist rate)
Amplitude
Time [s]
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
0.00200