Lab #4: Aliasing due to Undersampling

The first part of this lab is based upon the problems in Section 7.1, "Aliasing due to Undersampling," in the course text *Computer Explorations in SIGNALS AND SYSTEMS* by Buck, Daniel, Singer.

The second part relates to parts of Section 7.4, "Bandpass Sampling".

In this lab, you will explore the effect of aliasing due to undersampling on signals reconstructed by bandlimited interpolation. The lab makes use of the relationship between the CTFT of a periodic signal $x(t)$ and the DTFT of its samples $x[n]$, as well as the relationship between the DTFT of the samples $x[n]$ and the DTFS of $x[n]$.

In the following problems, you will examine the effects of undersampling a pure sinusoid and a chirp signal.

Problem 1: Sampling pure sinusoidal signals

Consider the sinusoidal signal

$$x(t) = sin(\Omega_0 t).$$

If $x(t)$ is sampled with frequency $\Omega_s = 2\pi/T$ rad/sec, then the discrete-time signal $x[n] = x(nT)$ is equal to

$$x[n] = sin(\Omega_0 nT).$$

Assume the sampling frequency is fixed at $\Omega_s = 2\pi(2048)$ rad/sec.

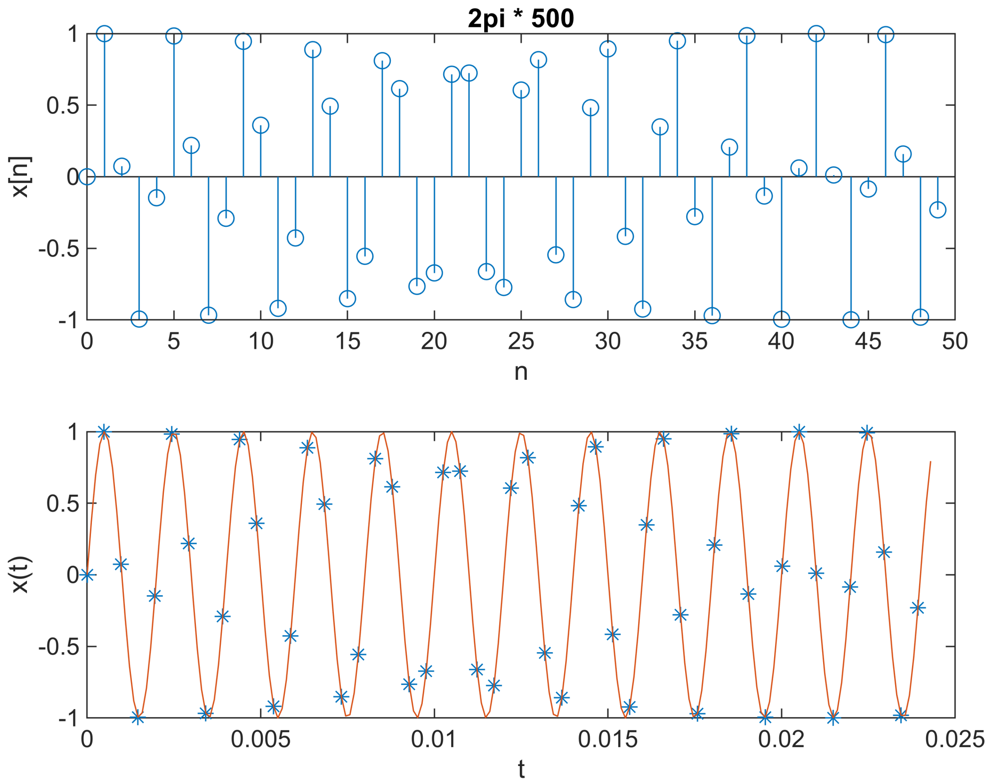
Task 1. Assume $\Omega_0 = 2\pi(500)$rad/sec and define `T=1/2048`. Create the vector `n=[0:2047]`, so that `t=n*T` contains 2048 time samples of the interval $0 \le t \le 1$. Create a vector x which contains the samples of $x(t)$ at the time samples in `t`. Then create a vector xs which contains 8192 samples of x(t) with spacing Ts=1/8192.

```
n = 0:2047;
T = 1/2048;
t = n*T;
omega_0 = 2*pi*500;
x = sin(omega_0*t);
ns=0:8191;
Ts=1/8192;
ts=ns*Ts;
xs = sin(omega_0*ts);
```

Task 2. Display the first 50 samples of x versus n using `stem`. Then plot the samples of x and xs on the same plot, overlaying the lower rate samples of x on the actual signal (represented by the higher sampling frequency of xs). (Use `subplot` to simultaneously display the two plots.)

```
figure;
subplot(2,1,1);
stem(n(1:50), x(1:50));
```

```
title('2pi * 500')
xlabel('n');
ylabel('x[n]');
subplot(2,1,2);
plot(t(1:50), x(1:50), '*');
hold on
plot(ts(1:200), xs(1:200));
xlabel('t');
ylabel('x(t)');
```



Note that `plot(ts,xs)` displays a continuous-time signal given the samples in x, using straight lines to interpolate between sample values. While this interpolation is not generally equal to the bandlimited reconstruction which follows from the sampling theorem, it can often be a very good approximation if the sampling rate is high enough.

To compute samples of the CTFT of the bandlimited reconstruction $x_r(t)$, we use the following function `ctfts`. It uses `fft` to calculate the Fourier transform of the reconstructed signal.

```
% function [X,f] = ctfts(x,T)
% CTFTS calculates the CTFT of a periodic signal x(t) which is reconstructed
% from the samples in the vector x using ideal bandlimited interpolation.
% the vector x contains samples x(t) over an integer number of periods, and
% T contains the sampling period.
%
% The vector X contains the area of the impulses at frequency values stored
```

```
% in the vector f.
%
% This function makes use of the relationship between the CTFT  of x(t) and
% the DTFT of its samples x[n], as well as the relationship between the
% DTFT of the samples x[n] and the DTFS of x[n].
% N=length(x);
% X=fftshift(fft(x,N))*(2*pi/N);
% f=linspace(-1,1-1/N,N)/(2*T);
%end
```

Task 3. Use `[X,f]=ctfts(x,T)` to calculate the CTFT of the reconstructed signal $x_r(t)$.

Write the analytical formula for $X(j\Omega)$m the CTFT of $x(t)$.

$X(j\Omega) = -j\pi(\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)) = j\pi\delta(\Omega + \Omega_0) - j\pi\delta(\Omega - \Omega_0)$.

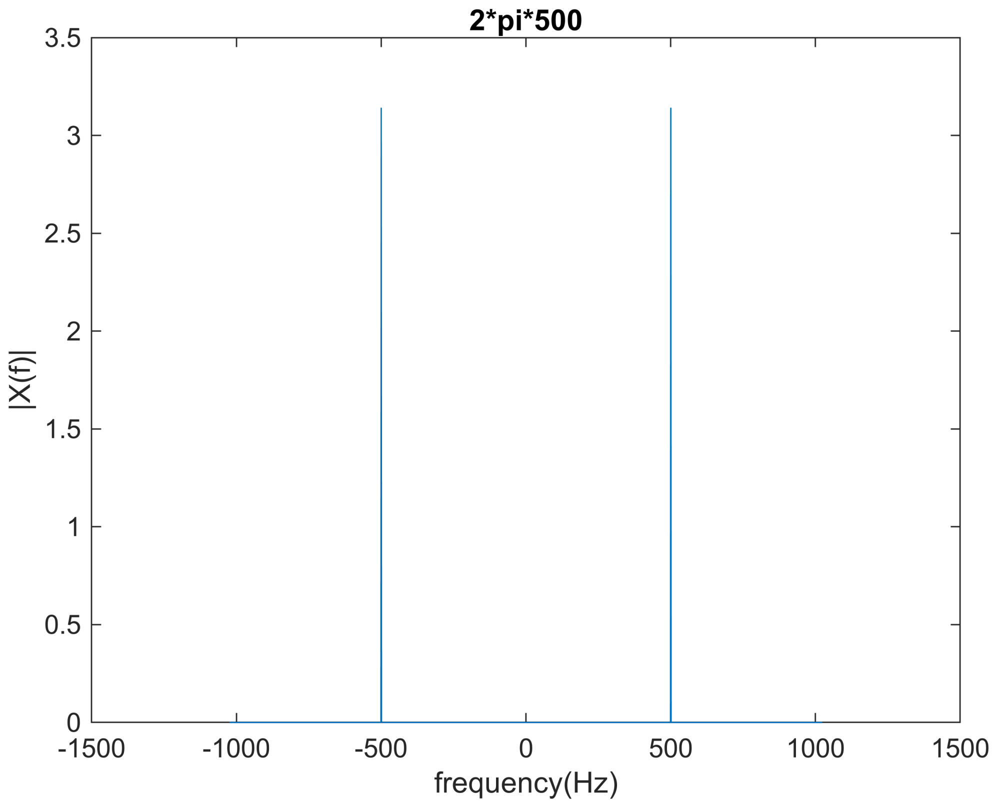
Plot the magnitude of `X` versus `f`.

Is `X` nonzero at the proper frequency values?

(Note that almost all of the elements in `X` are nonzero, but most are very small spurious values resulting from numerical round-off errors.)

X is nonzero and equal to π at $|f| = 500$ Hz, and close to zero otherwise (due to the numerical round-off errors in Matlab, the X vector has some small values that are close to zero).

```
[X,f] = ctfts(x, T);
figure;
plot(f, abs(X));
title('2*pi*500')
xlabel('frequency(Hz)');
ylabel('|X(f)|');
```

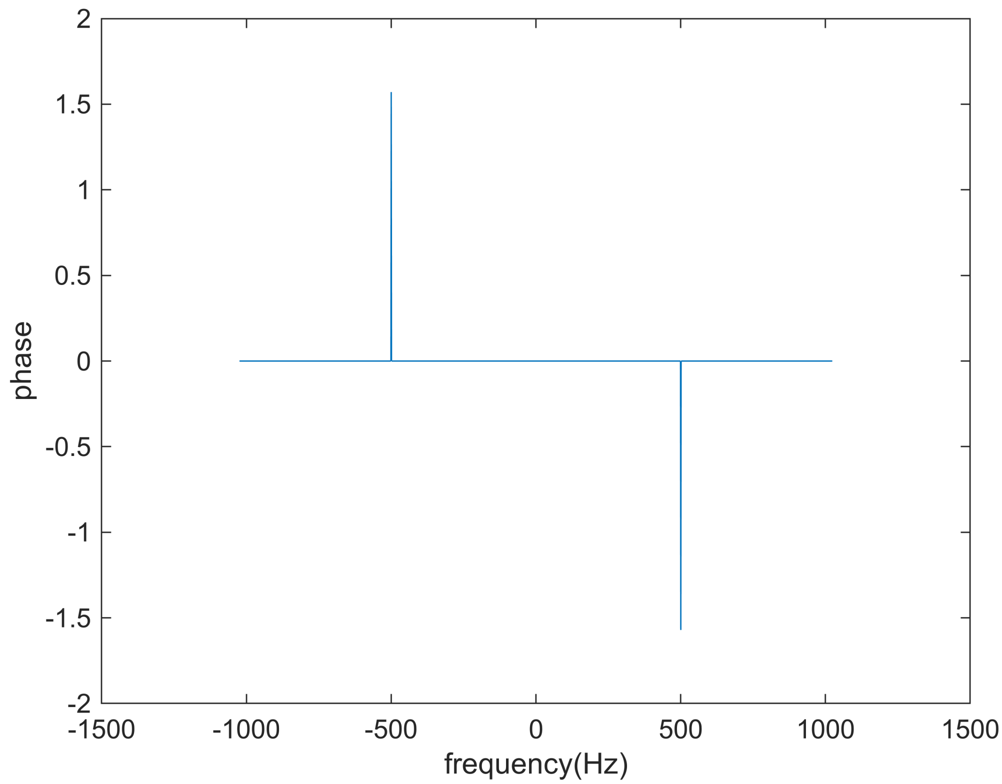


Task 4. Plot the phase of `X` versus `f`. (You can zero out the phase when the magnitude of X is small.)

Is the phase of `X` correct? (you can assume that the phase is equal to zero when `X` has a small non-zero value due to round-off error.)

The phase of X, after zeroing out the values with magnitude less than 0.25, has the correct phase, namely $-\pi/2$ at $f = 500$ Hz, and $\pi/2$ at $f = -500$ Hz.

```
figure;
plot(f, angle(X.*(abs(X)>0.25)));
xlabel('frequency(Hz)');
ylabel('phase');
```



You will now consider the effect of aliasing on the reconstructed signal when sampling pure sinusoidal signals of different frequencies.

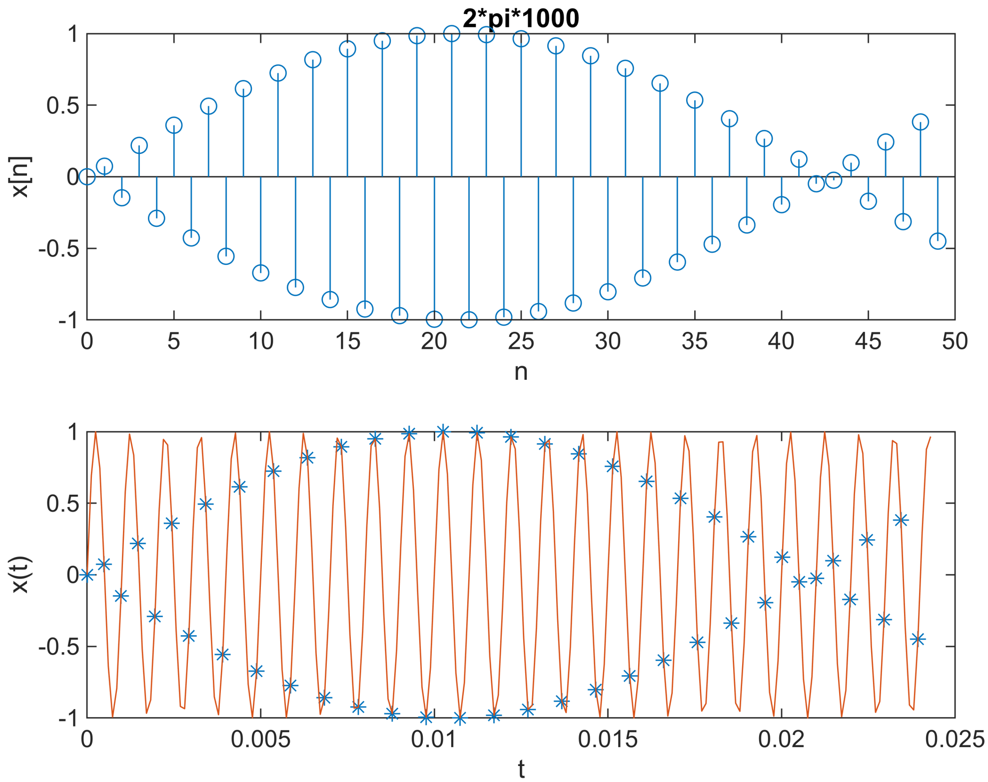
Task 5. Repeat Tasks 1 - 4 for the sinusoidal frequency $\Omega_0 = 2\pi(1000)$ rad/sec.

Again, is the magnitude of X nonzero for the expected frequencies? Is the phase of X correct?

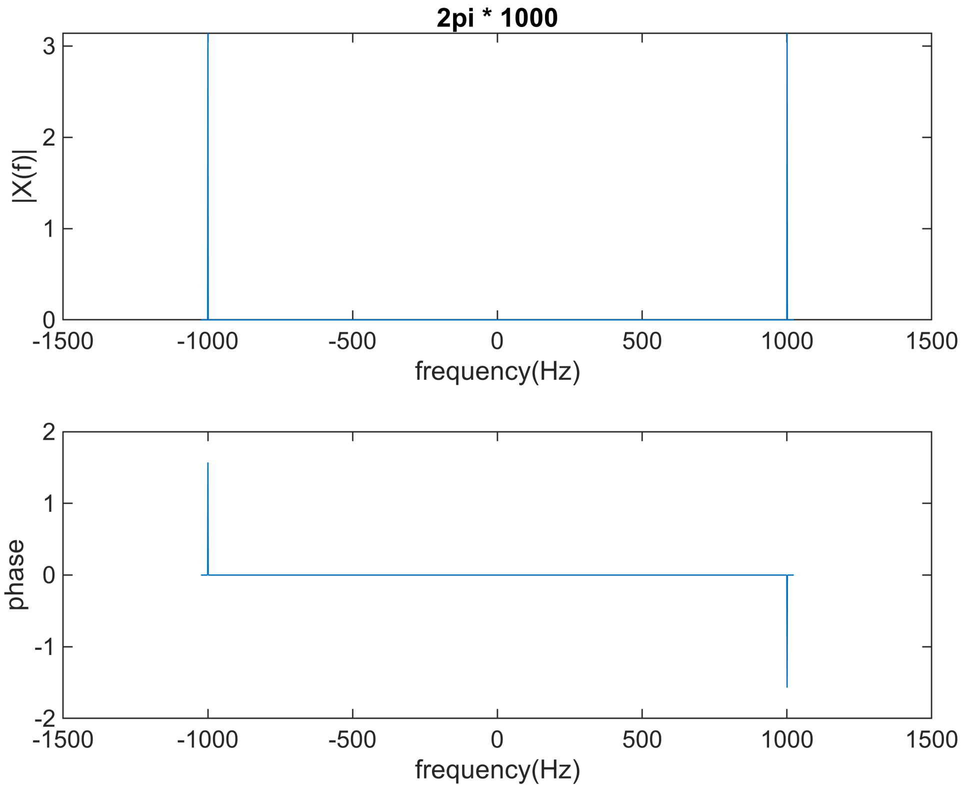
Yes, the magnitude and phase are what would be expected for these sinusoidal signals.

```
% Repeat for omega_0=2*pi*1000
n = 0:2047;
T = 1/2048;
t = n*T;
omega_0 = 2*pi*1000;
% Sample the signal and plot the first 50 samples as stem and line plots
x1 = sin(omega_0*t);
% Sample at the higher sampling frequency
ns=0:8191;
Ts=1/8192;
ts=ns*Ts;
xs1 = sin(omega_0*ts);
%
figure;
subplot(2,1,1);
stem(n(1:50), x1(1:50));
```

```
title('2*pi*1000')
xlabel('n');
ylabel('x[n]');
subplot(2,1,2);
plot(t(1:50), x1(1:50), '*');
hold on
plot(ts(1:200), xs1(1:200));
xlabel('t');
ylabel('x(t)');
```




```
% Compute and plot the CTFT magnitude and phase
[X,f] = ctfts(x1, T);
figure;
subplot(2,1,1);
plot(f, abs(X));
title('2pi * 1000')
xlabel('frequency(Hz)');
ylabel('|X(f)|');
subplot(2,1,2);
plot(f, angle(X.*(abs(X)>0.25)));
xlabel('frequency(Hz)');
ylabel('phase');
```



Task 6. Play the sampled signals created in Tasks 1 and 5 using sound(x,1/T). Does the pitch of the tone that you hear increase with increasing frequency Ω_0? Note that, like plot, the function sound performs interpolation. In essence, your computer converts the DT signal in MATLAB into a CT signal using a digital-to-analog converter, and then plays this CT signal on its speaker.

The pitch of tone increases with increasing frequency .

```
% Play sound for omega_0 = 2*pi*500;
omega_0 = 2*pi*500;
x = sin(omega_0*t);
sound(x, 1/T);
```

```
% Play sound for omega_0 = 2*pi*1000;
omega_0 = 2*pi*1000;
x1 = sin(omega_0*t);
sound(x1, 1/T);
```

Task 7. Now repeat Tasks 1,2, and 3 for the sinusoidal frequencies $\Omega_0 = 2\pi(1048)$ and $\Omega_0 = 2\pi(1548)$ rad/sec. Also play each sample signal using sound.

Does the pitch of the tone that you hear increase with each increase in the frequency Ω_0?

If not, can you explain the behavior?

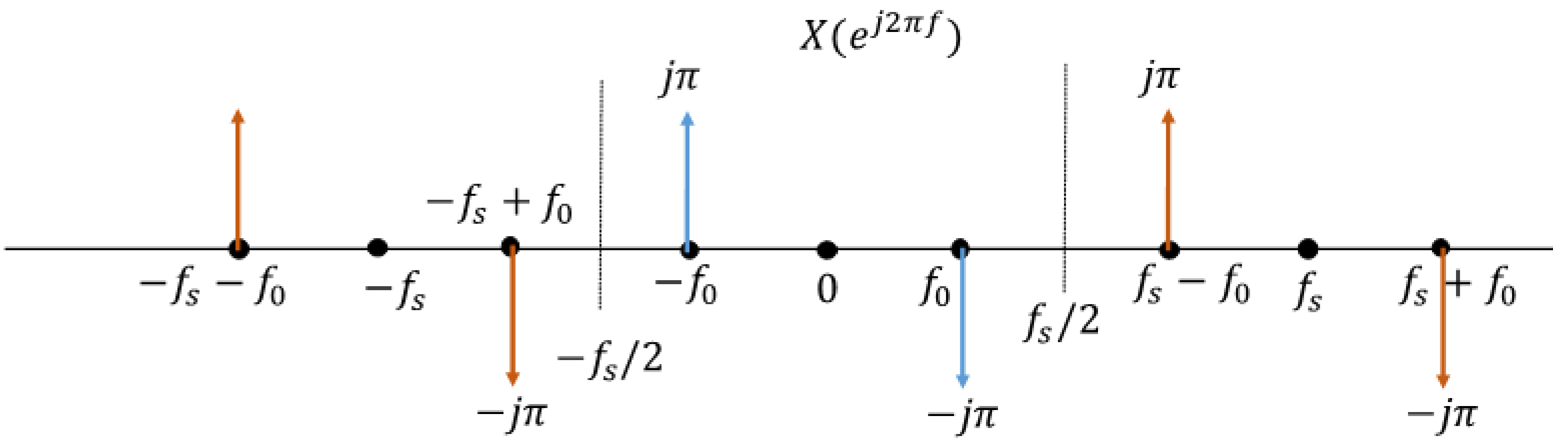
How do the tones for the cases $\Omega_0 = 2\pi(1048)$ and $\Omega_0 = 2\pi(1548)$ compare to the tones you heard for the cases $\Omega_0 = 2\pi(500)$ and $\Omega_0 = 2\pi(1000)$? Explain what you hear.

For $\Omega_0 = 2\pi(1048)$, X is nonzero at |f|=1048Hz. The phase is $\pi/2$ at f=1048Hz, and $-\pi/2$ at f= -1048Hz.

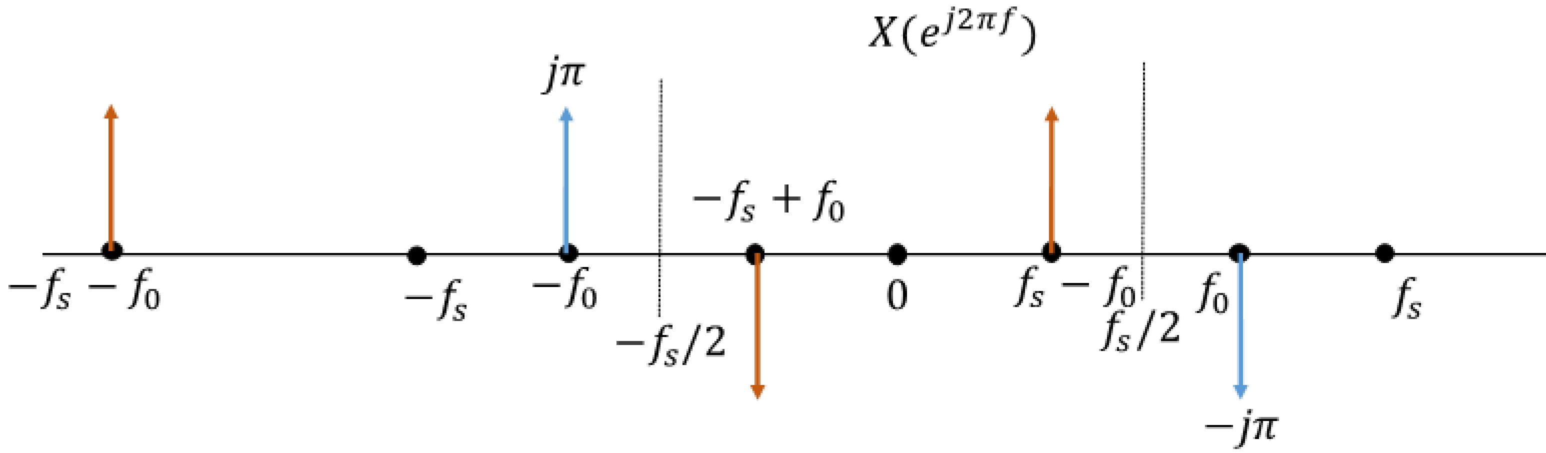
For $\Omega_0 = 2\pi(1548)$, X is nonzero at |f|=1548Hz. The phase is $\pi/2$ at f=1548Hz, and $-\pi/2$ at f= -1548Hz.

Playing these signals, we can notice that the pitch of tone decreases for $\Omega_0 = 2\pi(1048)$ rad/sec and $\Omega_0 = 2\pi(1548)$ rad/sec as the signal frequency nominally increases. Moreover, the pitch of the tone for $\Omega_0 = 2\pi(1048)$ rad/sec is identical to that of $\Omega_0 = 2\pi(1000)$, while the pitch for $\Omega_0 = 2\pi(1548)$ matches that of $\Omega_0 = 2\pi(500)$. We also observe that the phase is reversed (i.e., negated) for $\Omega_0 = 2\pi(1048)$ rad/sec and $\Omega_0 = 2\pi(1548)$ rad/sec compared to what would be expected from a sinusoidal signal.

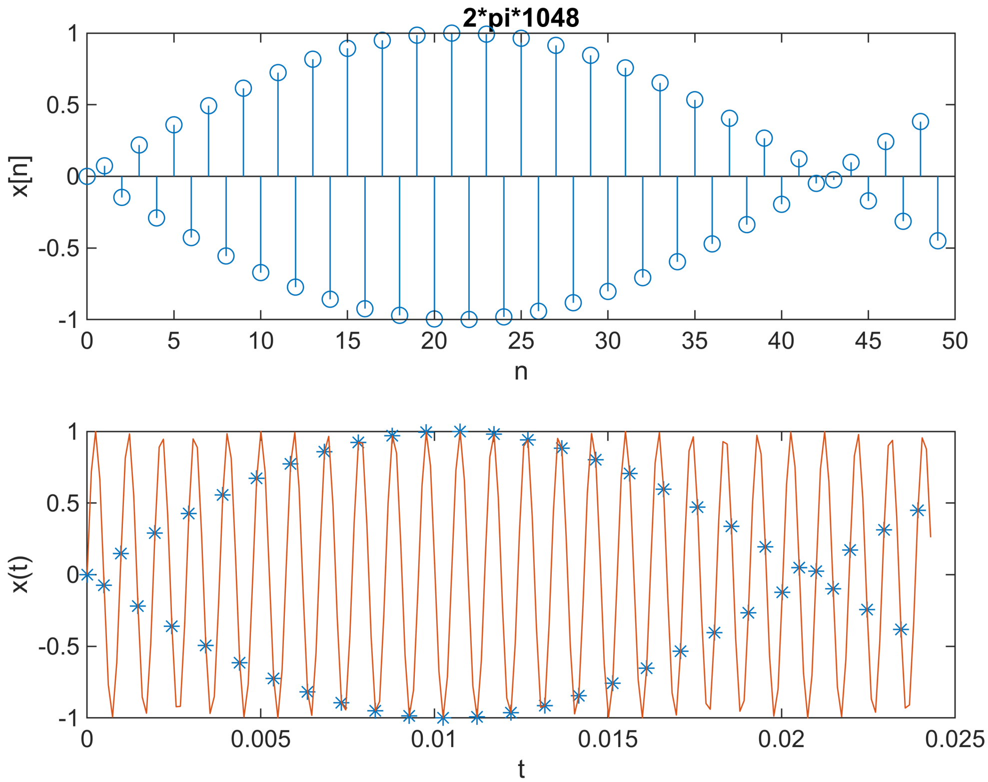
This behavior is because of aliasing. For $\Omega_0 < \Omega_s/2 = 2\pi(1024)$, the replicas of $X(j\Omega)$ do not overlap. No aliasing occurs. The vector X contains the spectrum within $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$. As Ω_0 increases, the pitch of the tone increases.



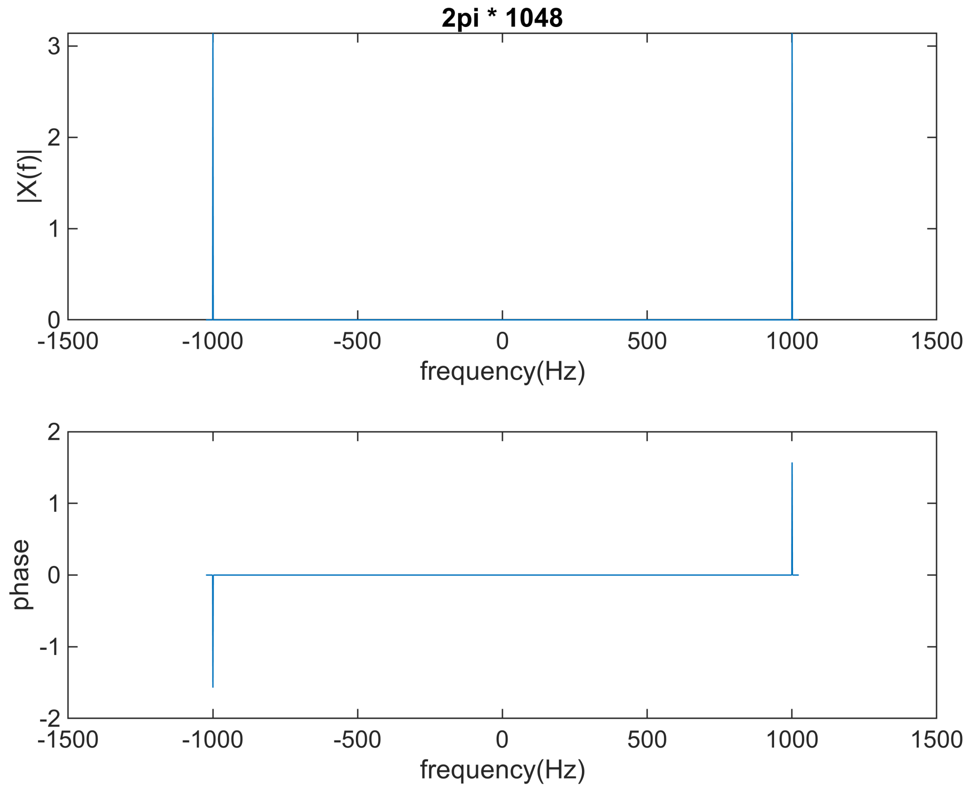
For $\Omega_0 < \Omega_s/2 = 2\pi(1024)$, aliasing occurs due to undersampling. The spectrum of the sampled signal is shown below. We can see that the vector X retained by `ctfts` consists of two impulses, each of which comes from one shifted replica of $X(j\Omega)$. As Ω_0 increases, these two impulses, which are colored red, become closer to zero frequency. Therefore, the pitch of tone we hear decreases. The tones for $\Omega_0 = 2\pi(1048)$ rad/sec and $\Omega_0 = 2\pi(1024)$ match because $f_s - f_0 = 2048 - 1048 = 1000$. Similarly, for $\Omega_0 = 2\pi(1548)$, we have $f_s - f_0 = 2048 - 1548 = 500$.




```
% For 2*pi*1048
%
omega_0 = 2*pi*1048;
x2 = sin(omega_0*t);
ns=0:8191;
Ts=1/8192;
ts=ns*Ts;
xs2 = sin(omega_0*ts);
%
figure;
subplot(2,1,1);
stem(n(1:50), x1(1:50));
title('2*pi*1048')
xlabel('n');
ylabel('x[n]');
subplot(2,1,2);
plot(t(1:50), x2(1:50), '*');
hold on
plot(ts(1:200), xs2(1:200));
xlabel('t');
ylabel('x(t)');
```

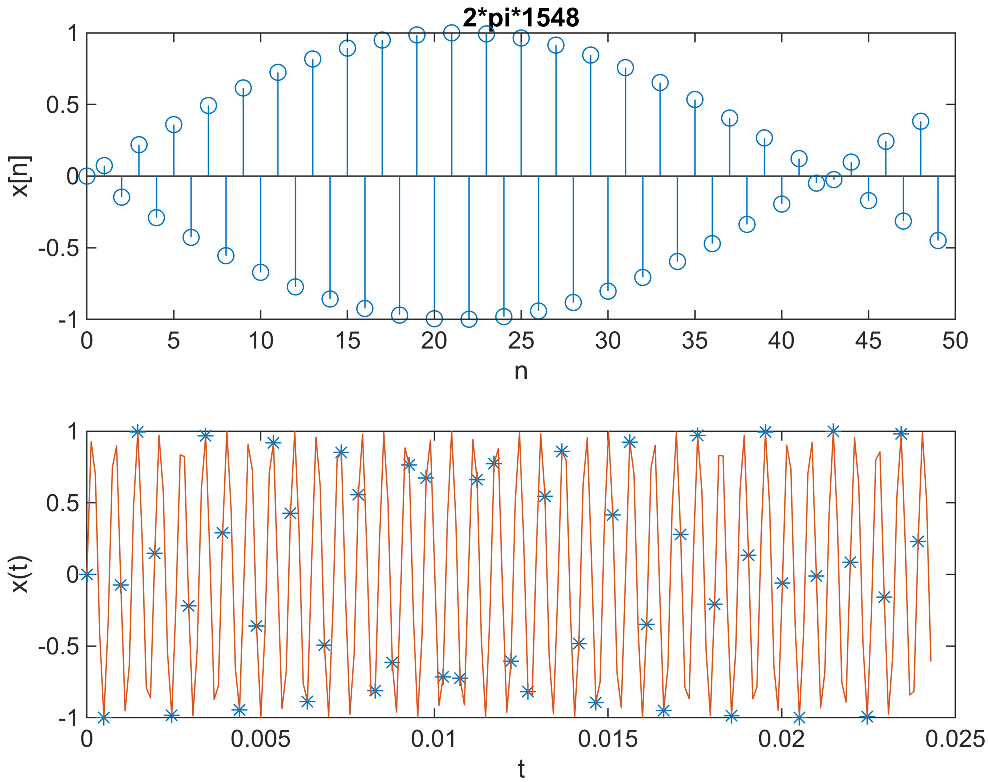



```
%
[X,f] = ctfts(x2, T);
figure;
subplot(2,1,1);
plot(f, abs(X));
title('2pi * 1048')
xlabel('frequency(Hz)');
ylabel('|X(f)|');
subplot(2,1,2);
plot(f, angle(X.*(abs(X)>0.25)));
xlabel('frequency(Hz)');
ylabel('phase');
```

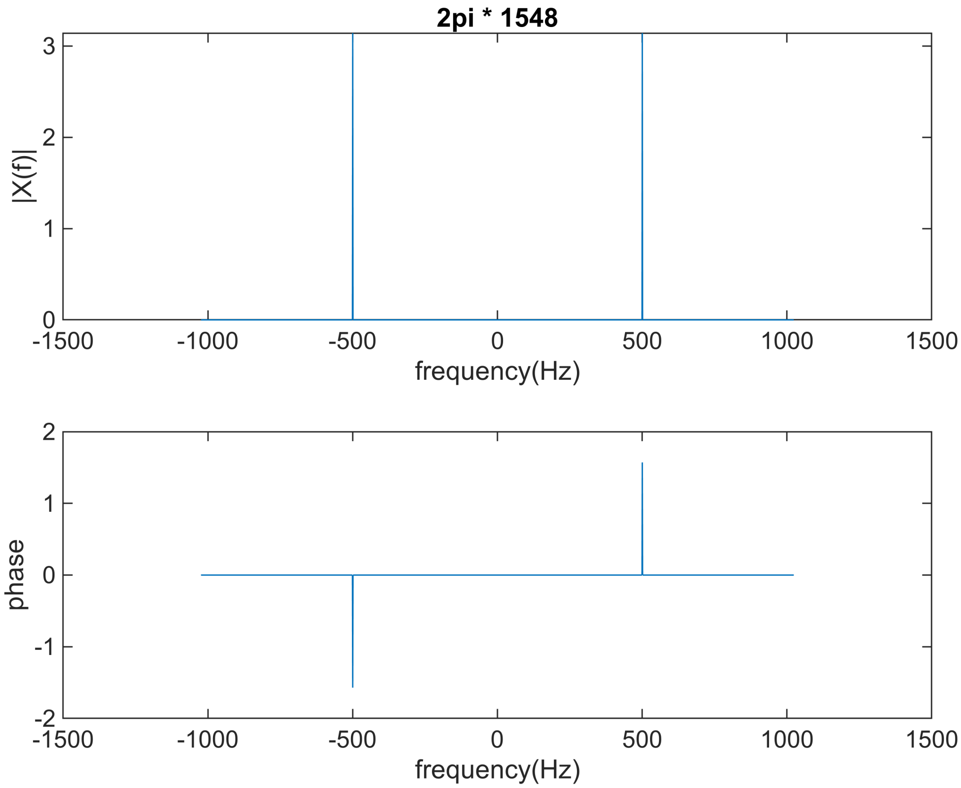



```
%
sound(x2, 1/T);
```

```
%  For 2*pi*1548
omega_0 = 2*pi*1548;
x3 = sin(omega_0*t);
ns=0:8191;
Ts=1/8192;
ts=ns*Ts;
xs3 = sin(omega_0*ts);
%
figure;
subplot(2,1,1);
stem(n(1:50), x1(1:50));
title('2*pi*1548')
xlabel('n');
ylabel('x[n]');
subplot(2,1,2);
plot(t(1:50), x3(1:50), '*');
hold on
plot(ts(1:200), xs3(1:200));
xlabel('t');
ylabel('x(t)');
```




```
%
[X,f] = ctfts(x3, T);
figure;
subplot(2,1,1);
plot(f, abs(X));
title('2pi * 1548')
xlabel('frequency(Hz)');
ylabel('|X(f)|');
subplot(2,1,2);
plot(f, angle(X.*(abs(X)>0.25)));
xlabel('frequency(Hz)');
ylabel('phase');
```




```
%
sound(x3, 1/T);
```

Problem 2: Sampling a chirp signal

Now consider the signal

$$x(t) = sin(\Omega_0 t + \frac{1}{2}\beta t^2),$$

which is often called a chirp signal due to the sound it makes when played through a loudpeaker. The "chirp" sound is due to the increasing instantaneous frequency of the signal over time. The instantaneous frequency of a sinusoidal signal is given by the derivative of its phase, i.e., the argument of $sin(\cdot)$. For the chirp signal, the instantaneous frequency is

$$\Omega_{\text{inst}} = \frac{d}{dt}\left(\Omega_0 t + \frac{1}{2}\beta t^2\right)$$
$$= \Omega_0 + \beta t.$$

For the following problems, set the sampling frequency to $\Omega_s = 2\pi(2048)$ rad/sec.

Task 1. Set $\Omega_0 = 2\pi(500)$ rad/sec and $\beta = 1000$ rad/sec^2. Store in the vector x the samples of the chirp signal on the interval $0 \leq t \leq 1$.

```
T = 1/2048;
t = (0:2047)*T;
omega_0 = 2*pi*500;
beta = 1000;
x = sin(omega_0*t+0.5*beta*t.^2);
```

Task 2. Use `sound` to play the chirp signal contained in `x1`. Qualitatively desribe what you just heard.

The signal has increasing instantaneous frequency, so the pitch continues to increase as a function of time.

```
sound(x, 1/T)
```

Task 3. Store in the vector `x6` the samples of the chirp signal on the interval $0 \leq t \leq 6$, and play the signal `x6` using `sound`. Determine (using a stopwatch) the approximate time instant at which the played signal has its maximum pitch (apparent frequency).

Given the linear equation for instantaneous frequency and your understanding of aliasing, explain how you could have predicted this time instant.

The instantaneous frequency at which the chirp signal has its maximum pitch satisfies $\Omega_{inst} = \Omega_0 + \beta t = \Omega_s/2$, which gives $t \approx 3.29239$ sec.

After that, aliasing begins to occur.

```
T = 1/2048;
t = (0:12287)*T;
omega_0 = 2*pi*500;
beta = 1000;
x6 = sin(omega_0*t+0.5*beta*t.^2);
%
sound(x6, 1/T);
```

Task 4. Store in the vector `x12` the samples of the chirp signal on the interval $0 \leq t \leq 12$, and play the signal `x12` using `sound`.

Determine (using a stopwatch) the approximate time sample at which the played chirp signal has zero (or very low) frequency.

Explain how you could have predicted this time sample.

The instantaneous frequency at which the chirp signal has its minimum pitch satisfies $\Omega_{inst} = \Omega_0 + \beta t = \Omega_s$, which gives $t \approx 9.72637$ sec.

Referring to the figure above, we see that when $\Omega_0 = \Omega_s$, the red impulses reach the lowest pitch, namely zero. They also cancel each other out at zero frequency, leading to zero volume.

```
T = 1/2048;
```

```
t = (0:24575)*T;
omega_0 = 2*pi*500;
beta = 1000;
x12 = sin(omega_0*t+0.5*beta*t.^2);
%
sound(x12, 1/T)
```

Problem 3: Bandpass sampling

This problem provides an introduction to the technique of bandpass sampling. When the signal of interest is a bandpass signal, there is a way to sample the signal below the Nyquist rate and still be able to fully reconstruct the signal. With an appropriately chosen sampling frequency, the periodically replicated copies of the signal spectrum in the spectrum of the sampled signal do not overlap, and the original signal can be recovered by means of a suitable recovery filter.

If the transform $X(j\Omega)$ of the bandpass signal $x(t)$ satisfies $X(j\Omega) = 0$ outside of the intervals $\Omega_L \le |\Omega| \le \Omega_H$, then a frequency Ω_s that satisfies the condition $\frac{2\Omega_H}{n} \le \Omega_s \le \frac{2\Omega_L}{n-1}$ for any integer $1 \le n \le \left\lfloor \frac{\Omega_H}{\Omega_H - \Omega_L} \right\rfloor$ will avoid aliasing and allow reconstruction of $x(t)$.

Task 1: Construct a chirp signal and show it is a bandpass signal. Specifically, consider the signal defined on the time interval $0 \le t \le t_0$ by $x(t) = \cos(\Omega_{min} t + \frac{\Omega_{max} - \Omega_{min}}{2t_0} t^2)$. Here we choose $\Omega_{min} = 2\pi(500)$, $\Omega_{max} = 2\pi(650)$, and $t_0 = 1$ second.

Create a vector x with 2048 samples of the chirp signal.

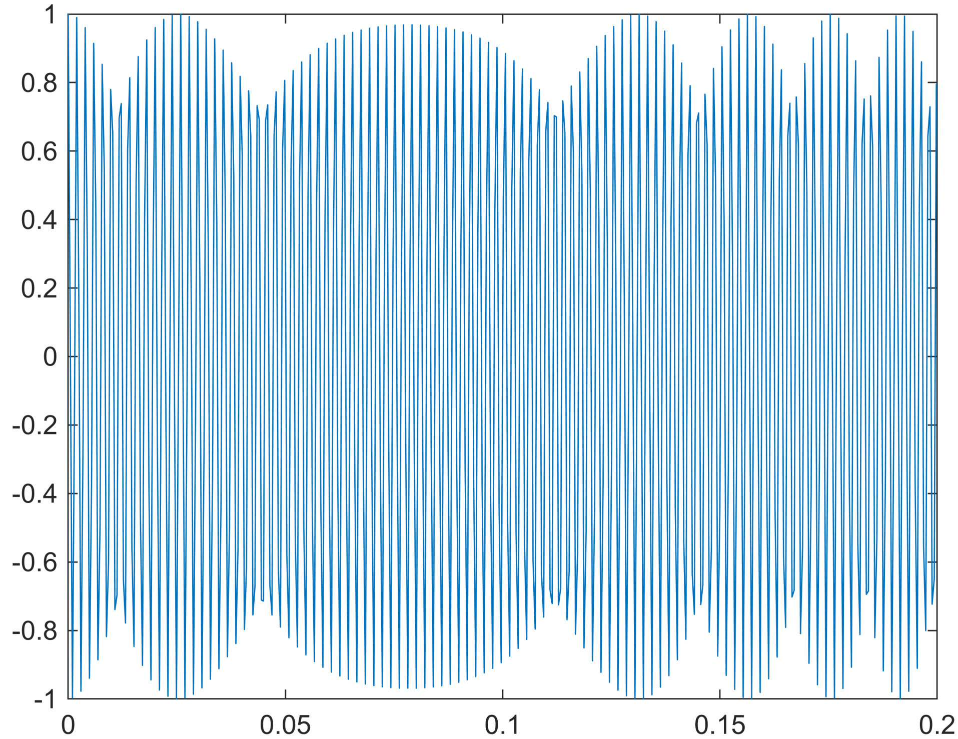
Plot about 1/5 second of the signal, i.e., 410 samples, against time (in seconds).

Listen to the signal x using the sound command.

<u>Does it sound qualitatively similar to the chirp signal you created in Problem 2?</u>

Yes, it sounds like a short segment of a chirp signal.

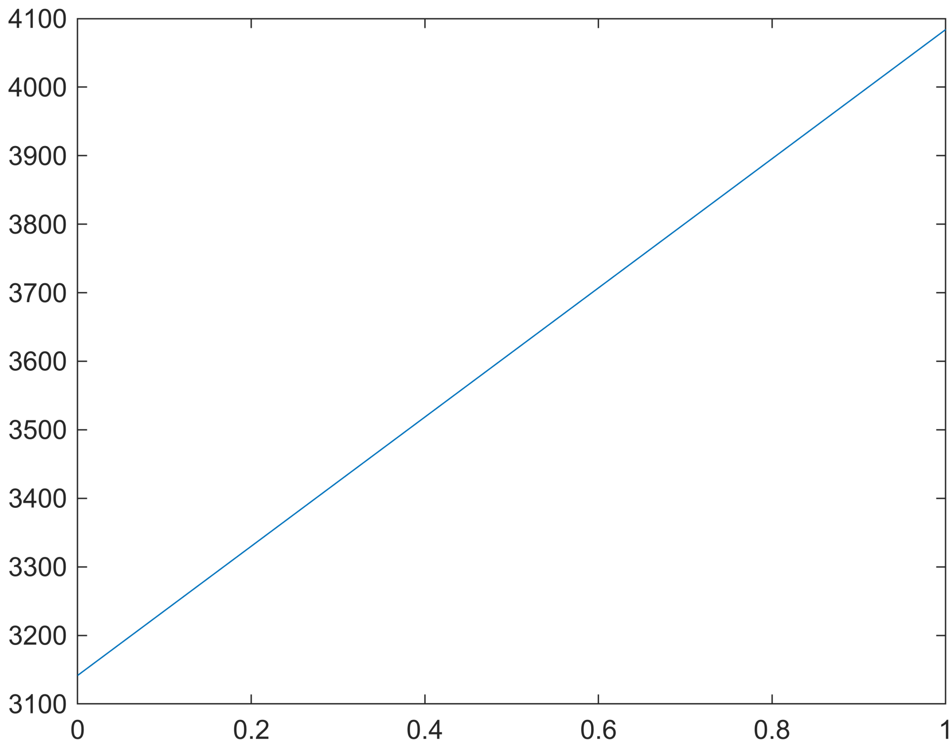
```
omega_min=2*pi*500;
omega_max=2*pi*650;
t0=1;
T =1/2048;
t =linspace(0,t0,t0/T);
x  = cos((omega_min*t)+((omega_max-omega_min)/(2*t0))*(t.^2));
figure;
plot(t(1:410),x(1:410));
```




```
%
sound(x,2048)
```

Task 2: Compute and plot the magnitude instantaneous frequency of the signal $x(t)$ against time (in seconds) over the interval $0 \le t \le 1$.

```
omega_inst=omega_min+((omega_max-omega_min)/t0)*t;
figure;
plot(t,omega_inst);
```



Task 3: Use the `fft` command to compute the Fourier transform of $x(t)$.

Plot it against frequency (in units of Hz).

Does the interval of non-zero transform values reflect the values of Ω_{min} and Ω_{max}?

We compute an M-point approximation to the Fourier Transform $X(j\omega)$ using the M-point fft of x , with a scale factor of $1/M$.

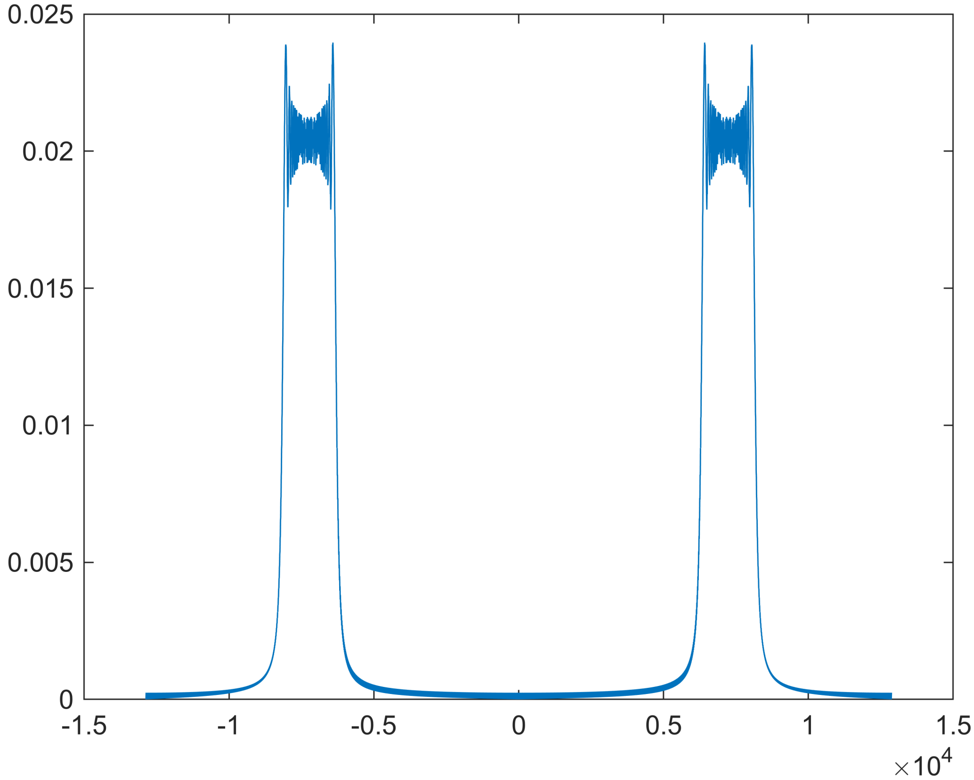
This approximation is justified by derivation of $X(j\omega)$ from the Fourier series coefficients of periodic replicas of x:

$X(jk\omega_0) = (1/M)a_k$

The interval of non-zero transform values is consistent with the instantaneous frequency interval $[500, 650]$ Hz.

```
M=4096;

T=1/M;
X=fftshift(fft(x,M))*(1/M);
f=2*pi*linspace((-1/(2*T)),(1/(2*T)),M);
figure;
plot(f,abs(X));
```



Task 4: You can use the formula in the introduction to this problem to calculate the allowable ranges of sampling frequencies that avoid aliasing and allow reconstruction of $x(t)$.

Write the value of n that gives the lowest possible interval of sampling frequencies.

n=4

Write the corresponding interval of sampling frequencies and the associated interval of sampling periods.

$[2\pi(1300/4), 2\pi(1000/3)] \approx [2\pi(325), 2\pi(333)]$

Task 5: The sampling frequency $\Omega_s = 2\pi(330)$ should fall among the sampling frequencies you found in Task 4.

Determine the frequency intervals within the range $[-\Omega_H, \Omega_H]$ where the copies of the spectrum of $x(t)$ in $[\Omega_L, \Omega_H]$ occur in the spectrum of the sampled signal.

In Hz: **[-490, -340]**, **[-160,-10]**, , **[170, 320]**, **[500,650]**

Do the same for the copies of the spectrum of $x(t)$ in $[-\Omega_H, -\Omega_L]$ Is there any overlap among these intervals?

In Hz: [-650, -500], [-320, -170], [10, 160], [340, 490]

There is no overlap, as can be seen by listing them in increasing order.

In Hz: [-650, -500], **[-490, -340]**, [-320, -170], **[-160,-10]**, [10, 160], **[170, 320]**, [340, 490], **[500,650]**

Task 6: Describe the reconstruction filter you would use to reconstruct $x(t)$ from the sampled sequence obtained with sampling frequency $\omega_s = 2\pi(330)$.

Use a bandpass filter with passband (in Hz) $[-650, -500] \cup [500, 650]$ and gain $T_s = \frac{1}{330}$.

```
0;
```

```
function [X,f] = ctfts(x,T)
% CTFTS calculates the CTFT of a periodic signal x(t) which is reconstructed
% from the samples in the vector x using ideal bandlimited interpolation.
% the vector x contains samples x(t) over an integer number of periods, and
% T contains the sampling period.
%
% The vector X contains the area of the impulses at frequency values stored
% in the vector f.
%
% This function makes use of the relationship between the CTFT  of x(t) and
% the DTFT of its samples x[n], as well as the relationship between the
% DTFT of the samples x[n] and the DTFS of x[n].
N=length(x);
X=fftshift(fft(x,N))*(2*pi/N);
f=linspace(-1,1-1/N,N)/(2*T);
end
```