Task 3: Signal Processing II

Name: Aleksandar Jakovljevic
Degree: BA
ID: 20613063

Problem 1

Part (a)

$$\begin{aligned} f(t) &= 5, |t| < 3 \\ g(t) &= 2, |t| < 3 \\ y(t) &= f(t) * g(t) \end{aligned}$$

$$y(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

We know that $y(t)$ is nonzero for $-6 < t < 6$, since both signals are $2 \times 3 = 6$ units wide.

$$\begin{aligned} &= \int_{-6}^{6} f(\tau)g(t-\tau)d\tau \\ &= f(\tau)\int_{-6}^{6} g(t-\tau)d\tau - \int_{-6}^{6} f'(\tau)\left(\int_{-6}^{6} g(t-\tau)\right)d\tau \end{aligned}$$

We know that $f(\tau) = 5$ where the function is nonzero, and since it is a constant then $f'(\tau) = 0$.

$$= 5\int_{-6}^{6} g(t-\tau)d\tau$$

We can now consider three cases: $-6 < t < 0$, $t = 0$, and $0 < t < 6$ over the bounds where we know the two functions interest ($f(t)$ is nonzero on the interval $-3 < t < 3$).

For $-6 < t < 0$:

$$\begin{aligned} &= 5\int_{-3-t}^{3} 2d\tau \\ &= 10\tau\Big|_{-3-t}^{3} \\ &= 10(3-(-3-t)) \\ &= 10(6+t) \\ &= 60 + 10t \end{aligned}$$

For $t = 0$:

$$\begin{aligned} &= 5\int_{-3}^{3} 2d\tau \\ &= 10\tau\Big|_{-3}^{3} \\ &= 10(3-(-3)) \\ &= 10(6) \\ &= 60 \end{aligned}$$

For $0 < t < 6$:

$$
\begin{aligned}
&= 5\int_{-3}^{3-t} 2d\tau \\
&= 10\tau\Big|_{-3}^{3-t} \\
&= 10((3-t)-(-3)) \\
&= 10(6-t) \\
&= 60-10t
\end{aligned}
$$

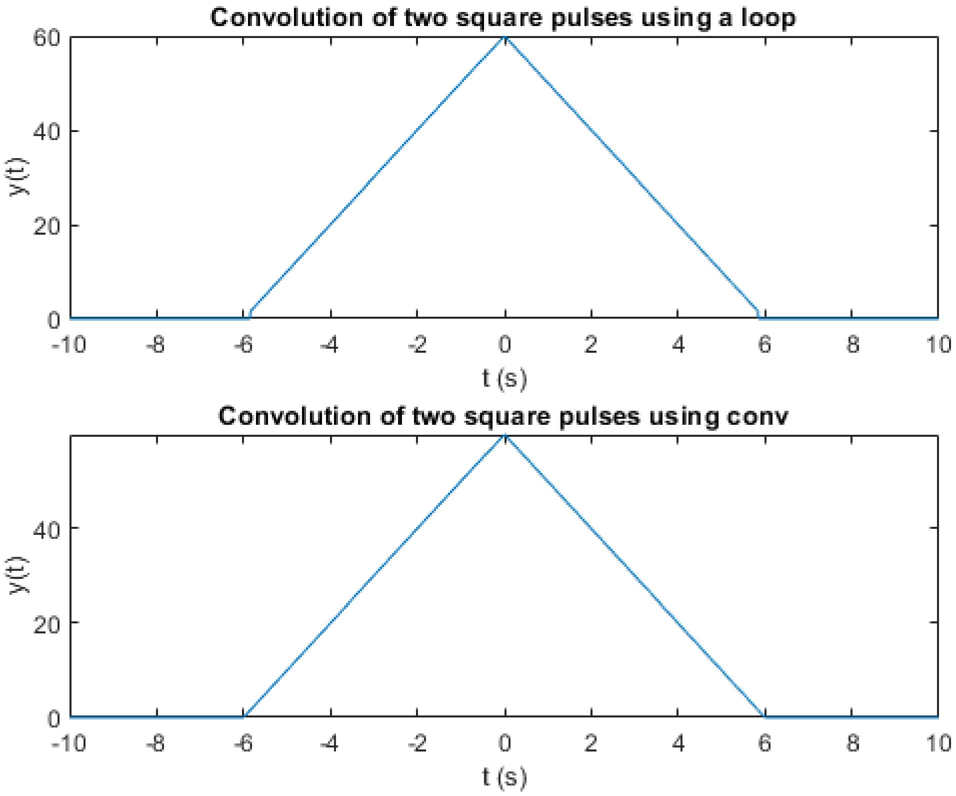
Therefore, $y(t)$ for the convolution of the two square pulses is calculated as:

$$
y(t) = \begin{Bmatrix} 60+10t & -6<t<0 \\ 60 & t=0 \\ 60-10t & 0<t<6 \\ 0 & \text{elsewhere} \end{Bmatrix}
$$

Parts (b) and (c)

```
% === 1b ================================================================ %

% setup parameters
dt = 0.01; % arbitrary
t_max = 10; % arbitrary
t1 = -t_max:dt:t_max;
A1 = 5;
a1 = 3;
B1 = 2;
syms f1(t) f2(t)
f1 = @(t) A1.*heaviside(a1 - abs(t));
g1 = @(t) B1.*heaviside(a1 - abs(t));
f2 = A1*heaviside(a1 - abs(t1));
g2 = B1*heaviside(a1 - abs(t1));

% perform convolution in a loop
y1b = zeros(1, length(t1));
count = 1;
for tau = t1
    fg1 = @(t) f1(t) .* g1(t - tau);
    y1b(count) = integral(fg1, -Inf, Inf);
    count = count + 1;
end

% plot
figure(1)
subplot(211)
plot(t1, y1b)
xlim([-t_max, t_max])
ylim([0, max(y1b)])
xlabel('t (s)')
ylabel('y(t)')
title('Convolution of two square pulses using a loop')

% === 1c ================================================================ %

% perform convolution using conv
subplot(212)
y1c = conv(f2, g2)*dt;
t1_mod = linspace(-2*t_max, 2*t_max, length(y1c));
```

```
plot(t1_mod, y1c)
xlim([-t_max, t_max])
ylim([0, max(y1c)])
xlabel('t (s)')
ylabel('y(t)')
title('Convolution of two square pulses using conv')
```



Problem 2

Part (a)

$$X(f) = \int_{-\infty}^{\infty} x(a) e^{i2\pi fa} da$$
$$H(f) = \int_{-\infty}^{\infty} h(b) e^{i2\pi fb} db$$

First theorem:

$$
\begin{aligned}
X(f)H(f) &= \int_{-\infty}^{\infty} x(\tau)e^{i2\pi f\tau}d\tau \int_{-\infty}^{\infty} h(t-\tau)e^{i2\pi f(t-\tau)}d(t-\tau) \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} h(t-\tau)e^{i2\pi f(t-\tau)}d(t-\tau)\right) x(\tau)e^{i2\pi f\tau}d\tau \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau\right) e^{i2\pi f(t-\tau)}e^{i2\pi f\tau}d(t-\tau) \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau\right) e^{i2\pi f(t-\tau)+i2\pi f\tau}d(t-\tau) \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau\right) e^{i2\pi ft}d(t-\tau) \\
&= F\left(\int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau\right) \\
&= F(x(t)*h(t))
\end{aligned}
$$

The above theorem demonstrates that multiplying together two Fourier-transformed functions in the frequency domain is the same as convolving them in the time domain and then Fourier transforming the result.

Second theorem:

$$
\begin{aligned}
X(f)*H(f) &= \int_{-\infty}^{\infty} x(t)e^{i2\pi ft}dt * \int_{-\infty}^{\infty} h(t)e^{i2\pi ft}dt \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} X(f_1)e^{i2\pi f_1 t}df_1\right)\left(\int_{-\infty}^{\infty} H(f-f_1)e^{i2\pi(f-f_1)t}d(f-f_1)\right)df \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(f-f_1)df_1 \int_{-\infty}^{\infty} X(f1)d(f-f_1)e^{i2\pi f_1 t}e^{i2\pi(f-f_1)t}df \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(f-f_1)df_1 \int_{-\infty}^{\infty} X(f1)d(f-f_1)e^{i2\pi f_1 t+i2\pi(f-f_1)t}df \\
&= \int_{-\infty}^{\infty} X(f)H(f)e^{i2\pi ft}df \\
&= F(x(t)h(t))
\end{aligned}
$$

The above theorem demonstrates that convolving together two Fourier-transformed functions in the frequency domain is the same as multiplying them in the time domain and then Fourier transforming the result.

Part (b)

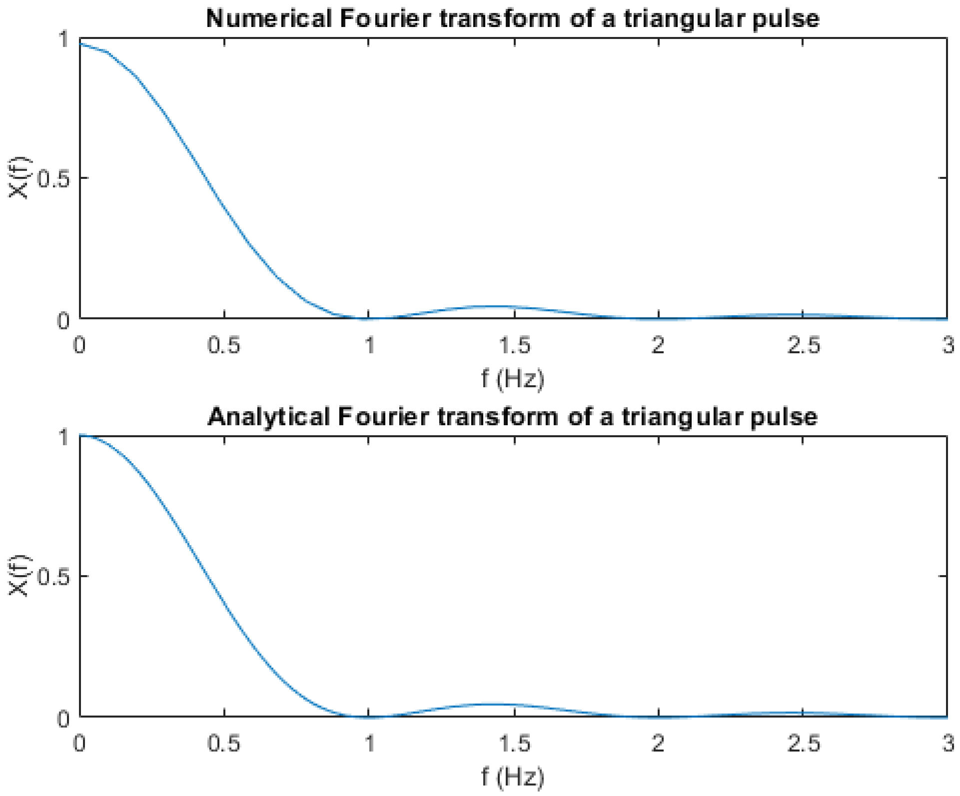
Analytic computation:

$$
\begin{aligned}
X(f) &= \int_{-\infty}^{\infty} x(t) e^{i2\pi ft} dt \\
&= \int_{-1}^{1} (1+|t|) e^{i2\pi ft} dt \\
&= \int_{-1}^{0} (1+t) e^{i2\pi ft} dt + \int_{0}^{1} (1-t) e^{i2\pi ft} dt \\
&= \int_{-1}^{0} e^{i2\pi ft} dt + \int_{-1}^{0} t e^{i2\pi ft} dt + \int_{0}^{1} e^{i2\pi ft} dt - \int_{0}^{1} t e^{i2\pi ft} dt \\
&= \int_{-1}^{1} e^{i2\pi ft} dt + \int_{-1}^{0} t e^{i2\pi ft} dt - \int_{0}^{1} t e^{i2\pi ft} dt \\
&= \left[\frac{e^{i2\pi ft}}{i2\pi f}\right]_{-1}^{1} + \left[\frac{i2\pi ft - 1}{(i2\pi f)^2} \times e^{i2\pi ft}\right]_{-1}^{0} - \left[\frac{i2\pi ft - 1}{(i2\pi f)^2} \times e^{i2\pi ft}\right]_{0}^{1} \\
&= \left[\frac{e^{i2\pi f}}{i2\pi f} - \frac{e^{-i2\pi f}}{i2\pi f}\right] + \left[\frac{-1}{(i2\pi f)^2} - \frac{i2\pi f + 1}{(i2\pi f)^2} \times e^{-i2\pi f}\right] - \left[\frac{i2\pi f - 1}{(i2\pi f)^2} \times e^{i2\pi f} + \frac{1}{(i2\pi f)^2}\right] \\
&= \frac{e^{i2\pi f} - e^{-i2\pi f}}{i2\pi f} - \frac{2}{(i2\pi f)^2} - \frac{e^{i2\pi f} - e^{-i2\pi f}}{i2\pi f} + \frac{e^{i2\pi f} + e^{-i2\pi f}}{(i2\pi f)^2} \\
&= \frac{1}{2\pi^2 f^2} - \frac{1}{2\pi^2 f^2} \times \frac{e^{i2\pi f} + e^{-i2\pi f}}{2} \\
&= \frac{1}{2\pi^2 f^2} (1 - \cos(2\pi f)) \\
&= \frac{1}{2\pi^2 f^2} (1 - (1 - 2\sin^2(\pi f)) \\
&= \frac{1}{2\pi^2 f^2} (2\sin^2(\pi f)) \\
&= \frac{\sin^2(\pi f)}{\pi^2 f^2} \\
&= \mathrm{sinc}^2(f)
\end{aligned}
$$

Numeric computation:

```
% === 2b ================================================================ %

% setup parameters
dt = 0.01; % arbitrary
fs = 100; % arbitrary
t_max = 3; % arbitrary
t2 = -t_max:dt:t_max;
n = 2^nextpow2(length(t2)); % improve fft accuracy

% compute Fourier transform numerically and compare to sinc^2 function
% as determined analytically
y2b = (1 - abs(t2)).*heaviside(1 - abs(t2));
y2b_n = fft(y2b, n);
f2 = fs*(0:(n/2))/n;
y2b_nP = abs(y2b_n/n)*log2(n);
y2b_a = (sinc(t2)).^2;

% plot
figure(2)
subplot(211)
plot(f2, y2b_nP(1:n/2+1))
xlim([0, 3])
ylim([0, 1])
xlabel('f (Hz)')
```

```
ylabel('X(f)')
title('Numerical Fourier transform of a triangular pulse')
subplot(212)
plot(t2, y2b_a)
xlim([0, 3])
ylim([0, 1])
xlabel('f (Hz)')
ylabel('X(f)')
title('Analytical Fourier transform of a triangular pulse')
```



Part (c)

In part (b), we found that the Fourier transform of a triangular pulse is $\mathrm{sinc}^2(f)$. We already know from previous knowledge that the Fourier transform of a square pulse is $\mathrm{sinc}(f)$. We also know from problem 1 that the convolution two square pulses is a triangular pulse.

Given what we proved in part (a):

$$F(x(t) * h(t)) = X(f)H(f)$$

We can therefore explain our result as:

$$F(square * square) = \mathrm{sinc} \times \mathrm{sinc}$$
$$F(triangle) = \mathrm{sinc}^2$$

Problem 3

Part (a)

The relationship shows that a discretized signal in the frequency domain, $X_s(f)$, is the same as the signal that has been shifted by some integer r multiple of the sampling frequency $1/\Delta$. As a result, it shows that $X_s(f) = X_s(f + r/\Delta)$. We already know this to be true in the time domain, as if we have a discretized periodic function $x_s(t)$ with a period of Δ, then shifting it by any integer multiple of the period will result in an identical signal.

Part (b)

The relationship shows that if we have a discretized signal $x_s(t)$ with a sampling frequency of $1/\Delta$ and we perform the Fourier transform on this signal, then the Fourier-transformed signal will be repeated at every integer multiple of $1/2\Delta$. This results in the wrap-around of adjacent repetitions of the signal in the frequency domain, and therefore (as seen in the graph) the actual signal cannot be accurately measured due to the aliasing that occurs around the Nyquist frequency.

Part (c)

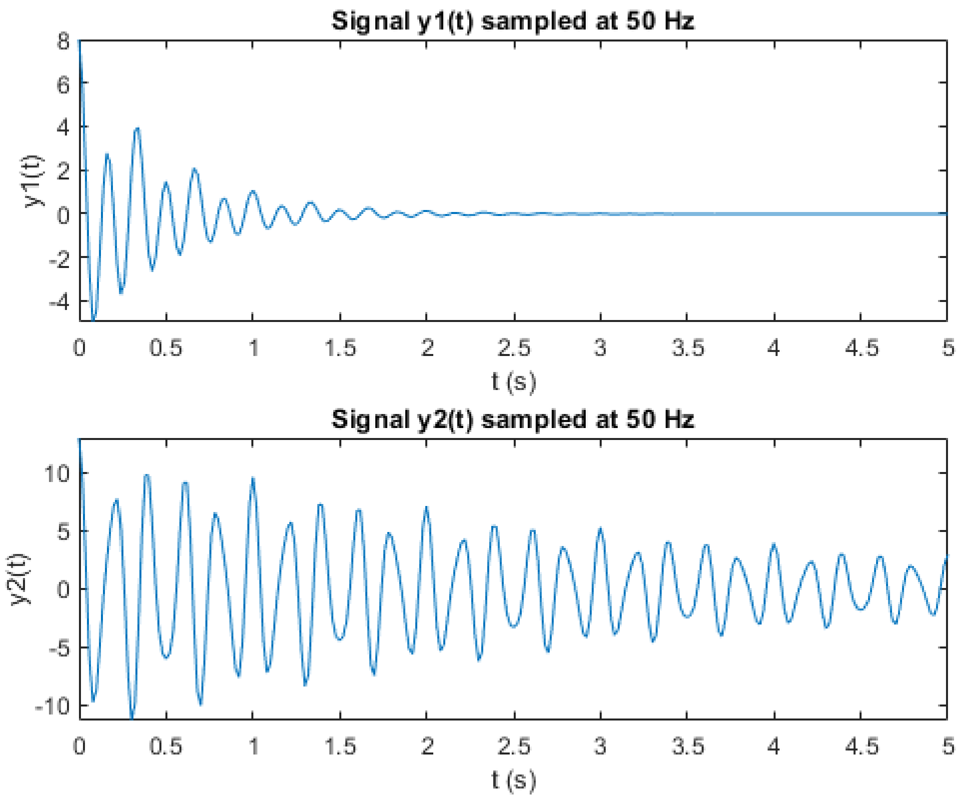
$X_s(f)$ is the Fourier transformation of the discretized signal $x_s(t)$ over the domain from $-\infty$ to ∞. $X_s(f)$ is a continuous function in the frequency domain. $X(k)$ however is not a continuous function, and is instead a finite sequence. $X(k)$ is obtained by sampling the original signal at some finite number of frequencies that are equally spaced apart.

Problem 4

Part (a)

```
% === 4a ================================================================ %

% setup parameters
a1 = 2;
b1 = 2;
c1 = 6;
f11 = 3;
f21 = 6;
a2 = 0.3;
b2 = 10;
c2 = 3;
f12 = 5;
f22 = 8;
t_max = 5;
% y1 = exp(-a1.*abs(t)).*((b1.*cos(2.*pi.*f11.*t))+(c1.*cos(2.*pi.*f21.*t)));
% y2 = exp(-a2.*abs(t)).*((b2.*cos(2.*pi.*f12.*t))+(c2.*cos(2.*pi.*f22.*t)));

% digitize signals
fs = 50;
ts = 0:1/fs:t_max;
z1 = exp(-a1.*abs(ts)).*((b1.*cos(2.*pi.*f11.*ts))+(c1.*cos(2.*pi.*f21.*ts)));
z2 = exp(-a2.*abs(ts)).*((b2.*cos(2.*pi.*f12.*ts))+(c2.*cos(2.*pi.*f22.*ts)));

% plot
figure(3)
subplot(211)
```

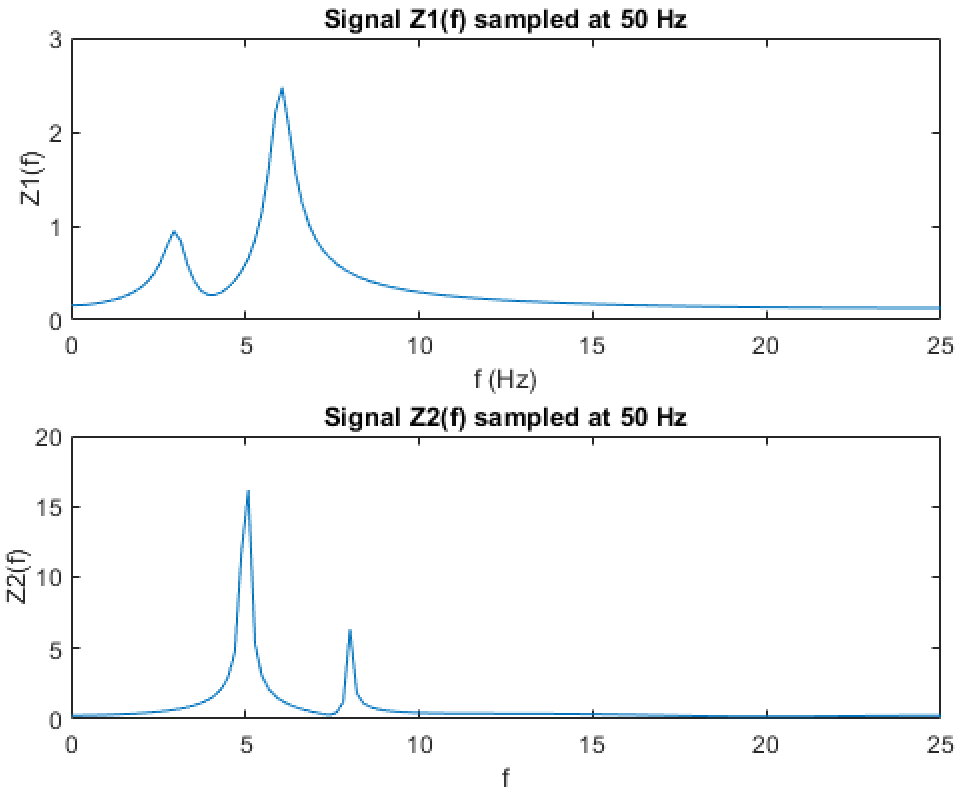
```
plot(ts, z1)
xlim([0, t_max])
ylim([min(z1), max(z1)])
xlabel('t (s)')
ylabel('y1(t)')
title('Signal y1(t) sampled at 50 Hz')
subplot(212)
plot(ts, z2)
xlim([0, t_max])
ylim([min(z2), max(z2)])
xlabel('t (s)')
ylabel('y2(t)')
title('Signal y2(t) sampled at 50 Hz')
```



Part (b)

```
% === 4b ============================================================== %

% compute Fourier transform
n = 2^nextpow2(length(ts)); % improve fft accuracy
z1_ft = fft(z1, n);
z2_ft = fft(z2, n);
ff = fs*(0:(n/2))/n;
z1_ftP = abs(z1_ft/n)*log2(n);
z2_ftP = abs(z2_ft/n)*log2(n);

% plot
figure(4)
subplot(211)
plot(ff, z1_ftP(1:n/2+1))
xlabel('f (Hz)')
```

```
ylabel('Z1(f)')
title('Signal Z1(f) sampled at 50 Hz')
subplot(212)
plot(ff, z2_ftP(1:n/2+1))
xlabel('f')
ylabel('Z2(f)')
title('Signal Z2(f) sampled at 50 Hz')
```



Part (c)

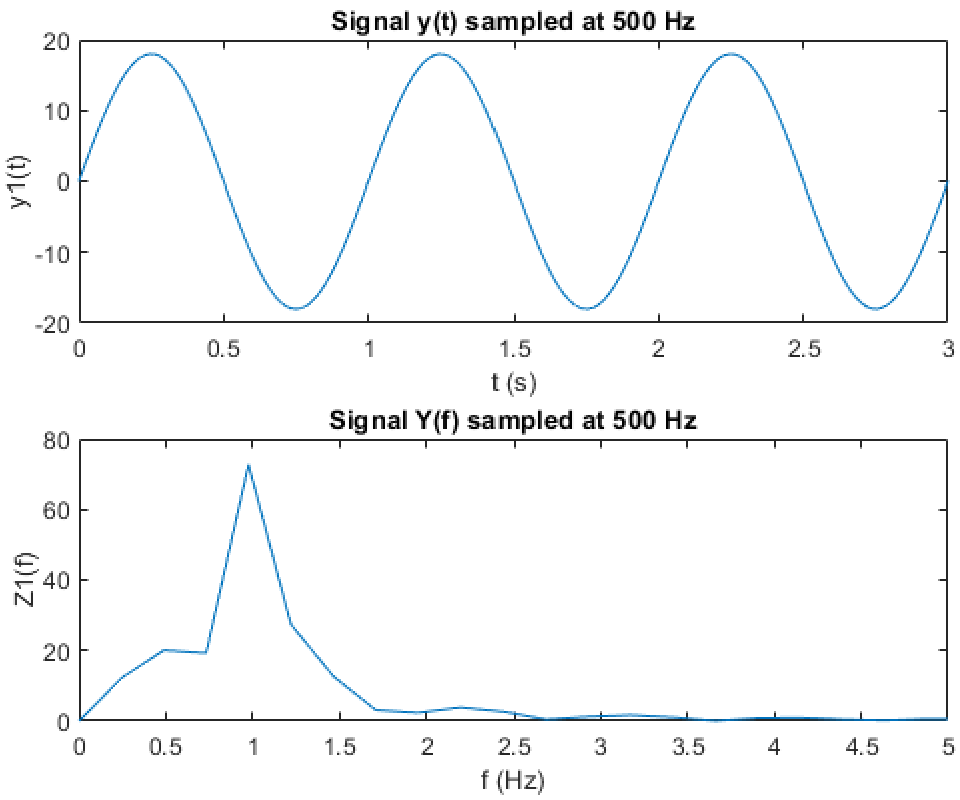
The frequency curve of z_2 is more narrow as it has a higher amplitude at the peak frequencies, and these peaks are spread over a smaller interval of values of frequency. The reason for this difference is that y_2 is "wider" in the time domain due to its lower value of a, which causes the exponential function to have a lesser slope than the exponential function in y_1.

Problem 5

Parts (a and b)

```
% === 5a ============================================================= %

a1 = 3;
a2 = 10;
a3 = 5;
fs = 500;
t_max = 3;
ts = 0:1/fs:t_max;
y1 = (a1*sin(2*pi*ts))+(a2*sin(2*pi*ts))+(a3*sin(2*pi*ts));
```

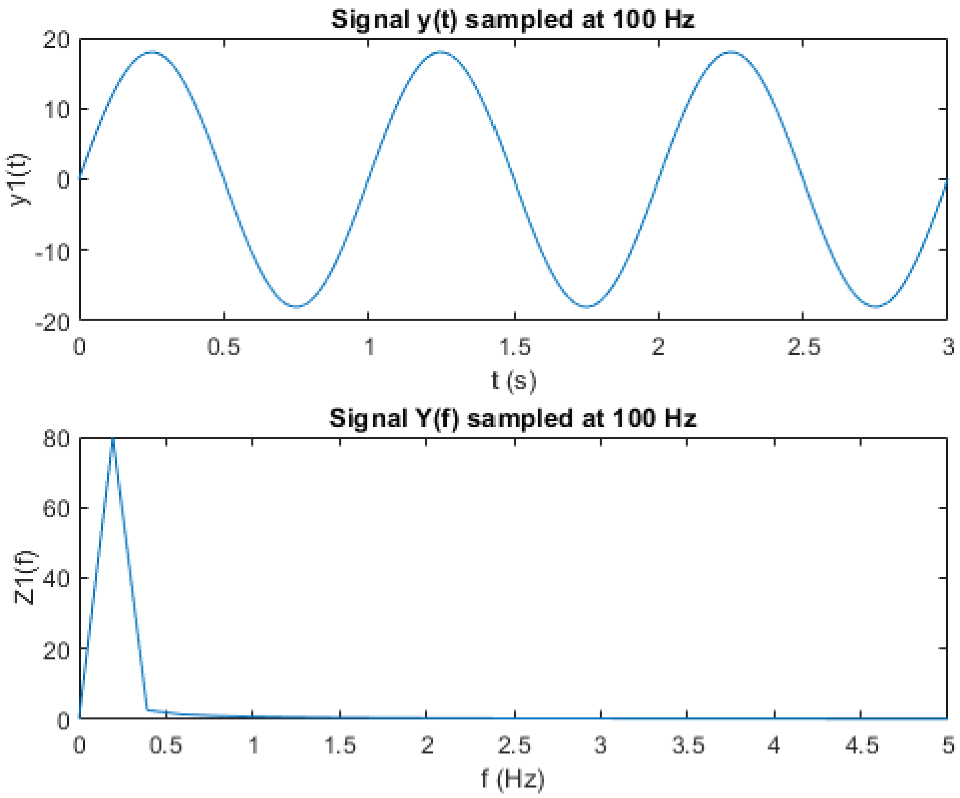
```
figure(5)
subplot(211)
plot(ts, y1);
xlabel('t (s)')
ylabel('y1(t)')
title('Signal y(t) sampled at 500 Hz')

% === 5b ============================================================== %

n = 2^nextpow2(length(ts)); % improve fft accuracy
y1_ft = fft(y1, n);
ff = fs*(0:(n/2))/n;
y1_ftP = abs(y1_ft/n)*log2(n);

% plot
subplot(212)
plot(ff, y1_ftP(1:n/2+1))
xlim([0, 5])
xlabel('f (Hz)')
ylabel('Z1(f)')
title('Signal Y(f) sampled at 500 Hz')
```



Parts (c and d)

```
% === 5c ============================================================== %

fs = 100;
ts = 0:1/fs:t_max;
y2 = (a1*sin(2*pi*ts))+(a2*sin(2*pi*ts))+(a3*sin(2*pi*ts));

figure(6)
```

```
subplot(211)
plot(ts, y2);
xlabel('t (s)')
ylabel('y1(t)')
title('Signal y(t) sampled at 100 Hz')

% === 5d ================================================================ %

n = 2^nextpow2(length(ts)); % improve fft accuracy
y2_ft = fft(y1, n);
ff = fs*(0:(n/2))/n;
y2_ftP = abs(y2_ft/n)*log2(n);

% plot
subplot(212)
plot(ff, y2_ftP(1:n/2+1))
xlim([0, 5])
xlabel('f (Hz)')
ylabel('Z1(f)')
title('Signal Y(f) sampled at 100 Hz')
```



Part (e)

No, all the original frequencies in $y(t)$ cannot be measured with a sampling rate of 100 Hz regardless of sampling duration, since the highest frequency in the original signal (125 Hz) is higher than the Nyquist frequency (50 Hz).

Part (f)

No, all the original frequencies in $y(t)$ cannot be measured with a sampling rate of 105 Hz, since the highest frequency in the original signal (125 Hz) is higher than the Nyquist frequency (52.5 Hz).

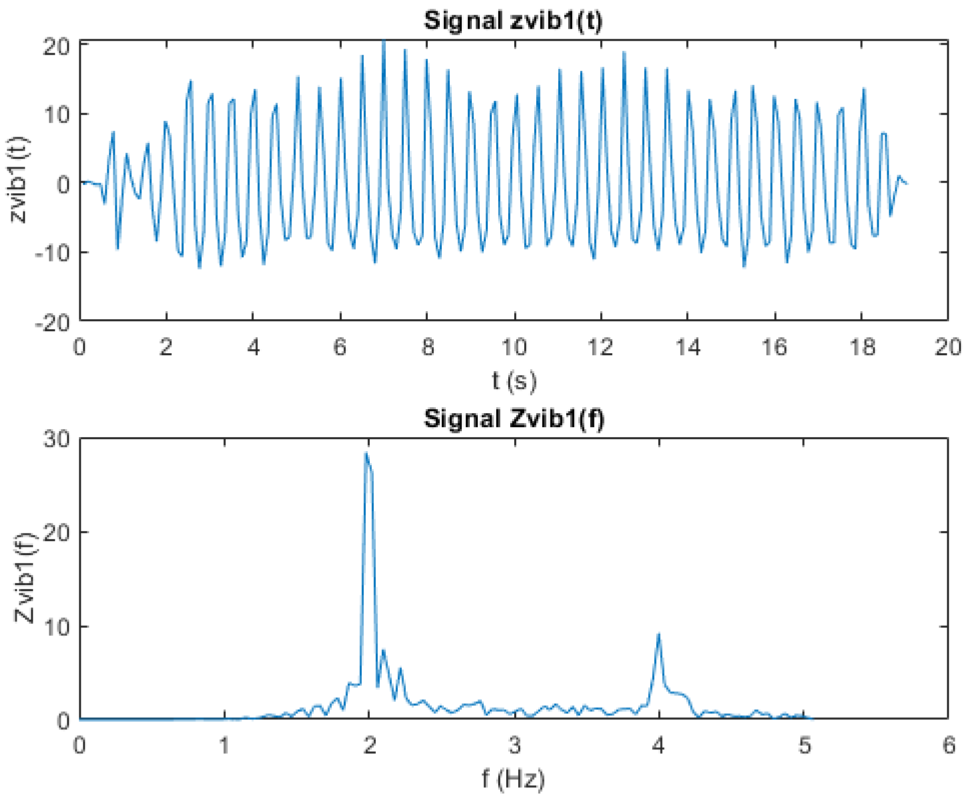
Part (g)

Despite the fact that with a sampling frequency of 251 Hz we have a Nyquist frequency (125.5 Hz) that is higher than the highest frequency in the original signal (125 Hz), the between these two is very small. Given the sampling duration of only three seconds, this is probably too short to accurately capture enough information about the signal to extract all of the frequencies in the original signal.

Problem 6

Part (a)

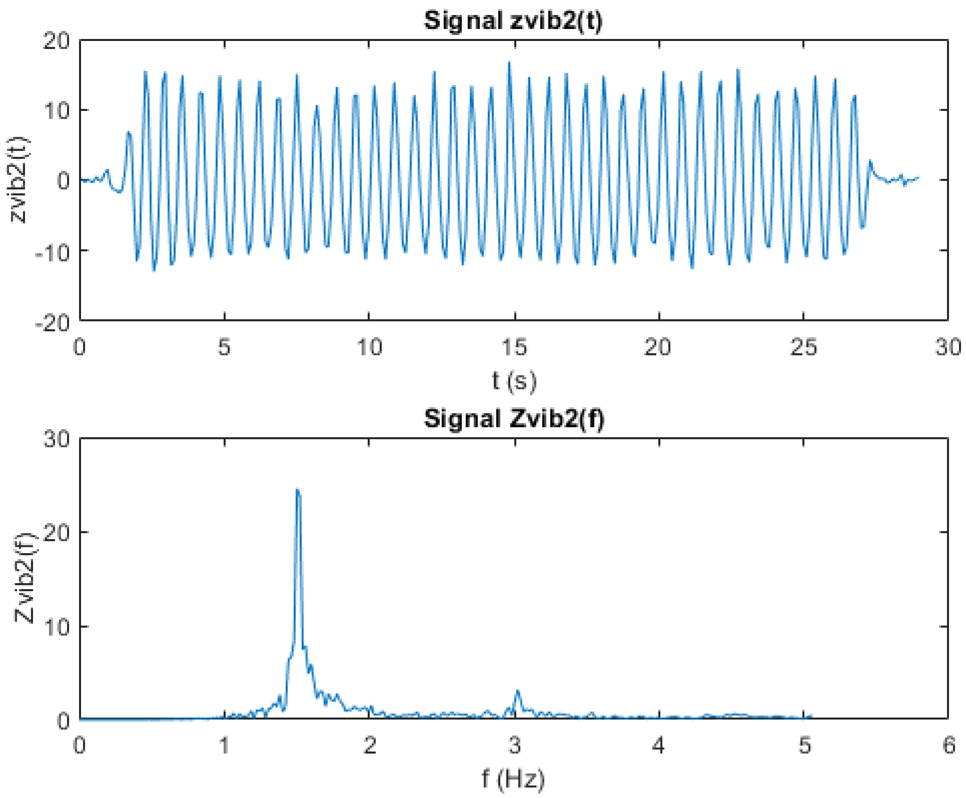
```
% === 6a ================================================================ %

load('data1.mat')
ts = 1:length(zvib);
ts = ts/fs;

figure(7)
subplot(211)
plot(ts, zvib);
xlabel('t (s)')
ylabel('zvib1(t)')
title('Signal zvib1(t)')

n = 2^nextpow2(length(ts)); % improve fft accuracy
zvib_ft = fft(zvib, n);
ff = fs*(0:(n/2))/n;
zvib_ftP = abs(zvib_ft/n)*log2(n);

subplot(212)
plot(ff, zvib_ftP(1:n/2+1))
xlabel('f (Hz)')
ylabel('Zvib1(f)')
title('Signal Zvib1(f)')
```



From the above plot, we can see that the main frequency of this signal is 2 Hz.

Part (b)

```
% === 6b ============================================================== %

load('data2.mat')
ts = 1:length(zvib);
ts = ts/fs;

figure(8)
subplot(211)
plot(ts, zvib);
xlabel('t (s)')
ylabel('zvib2(t)')
title('Signal zvib2(t)')

n = 2^nextpow2(length(ts)); % improve fft accuracy
zvib_ft = fft(zvib, n);
ff = fs*(0:(n/2))/n;
zvib_ftP = abs(zvib_ft/n)*log2(n);

subplot(212)
plot(ff, zvib_ftP(1:n/2+1))
xlabel('f (Hz)')
ylabel('Zvib2(f)')
title('Signal zvib2(f)')
```



From the above plot, we can see that the main frequency of this signal is 1.5 Hz.