

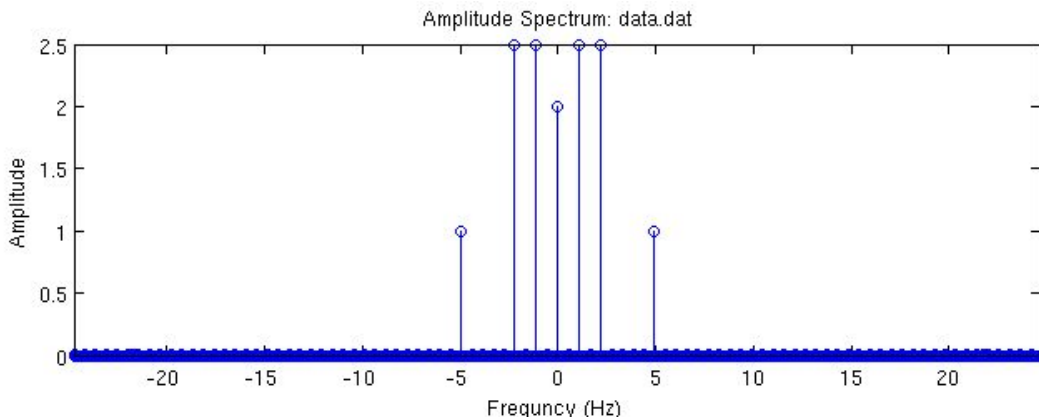
# **Lab 4 - Fourier Analysis - GPGN 404**

**23 October 2015**

**Garrett F. Sickles**

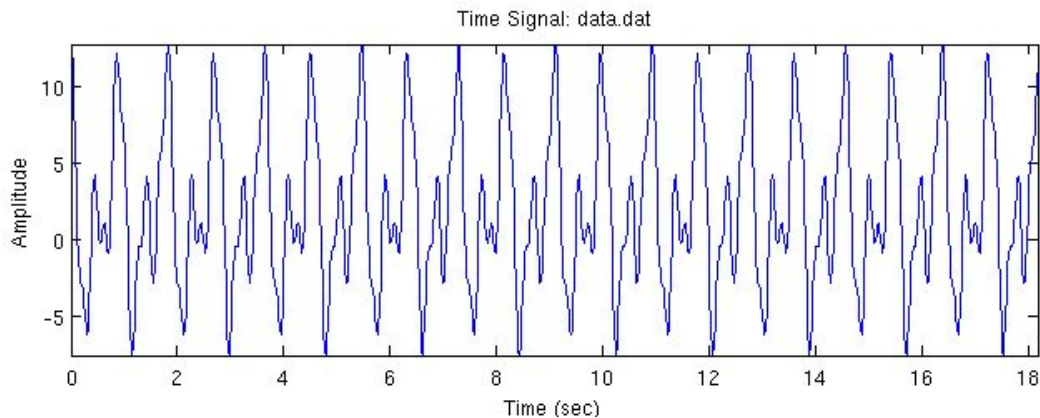
1) Using the time series in the file *Lab4\_t\_xt.dat*, answer the following questions:

- a) The sampling frequency is  $\frac{900 \text{ Samples}}{18.16162 \text{ sec}} = 49.5555 \text{ Hz}$ . However upon further inspection it can be deduced the amount of time per sampling interval is actually at least  $0.02020202 \text{ sec}$ . Every fifth sample in the data increases by at least  $0.00021 \text{ sec}$  and every hundred samples this increase only takes four samples to occur. This means a more accurate sampling frequency could be  $\frac{1}{0.02020202 \text{ sec}} = 49.50000 \text{ Hz}$ .
- b) The DC component of the signal is 2.



**Figure 1:** Amplitude Spectrum of the time signal in *data.dat*

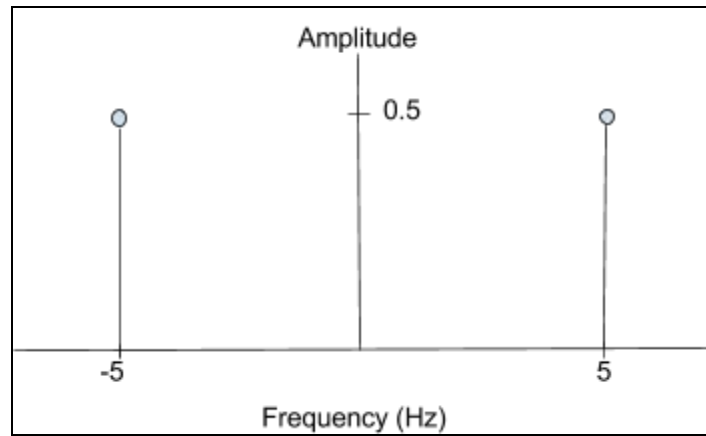
- c) By inspection the time period of this signal is 1.818 seconds.



**Figure 2:** Plot of the time signal in *data.dat*

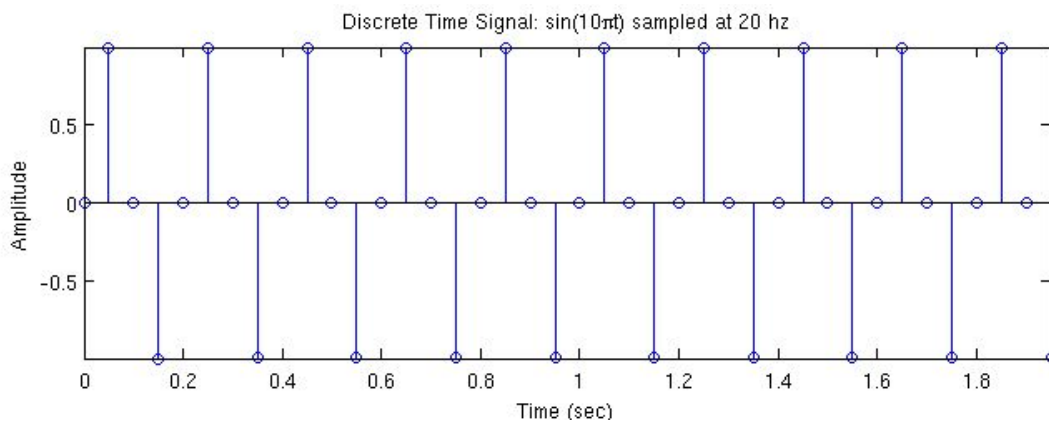
2) Aliasing and the Nyquist Frequency for the time signal  $x(t) = \sin(10\pi t)$

a) Sketch the two-sided amplitude spectrum  $F[x(t)]$

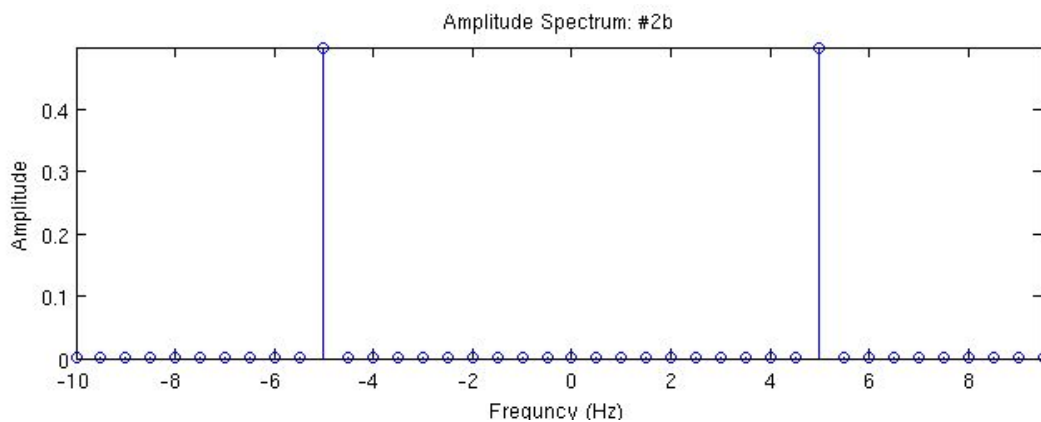


**Figure 3:** Sketch of the two sided amplitude spectrum for the time signal in number two

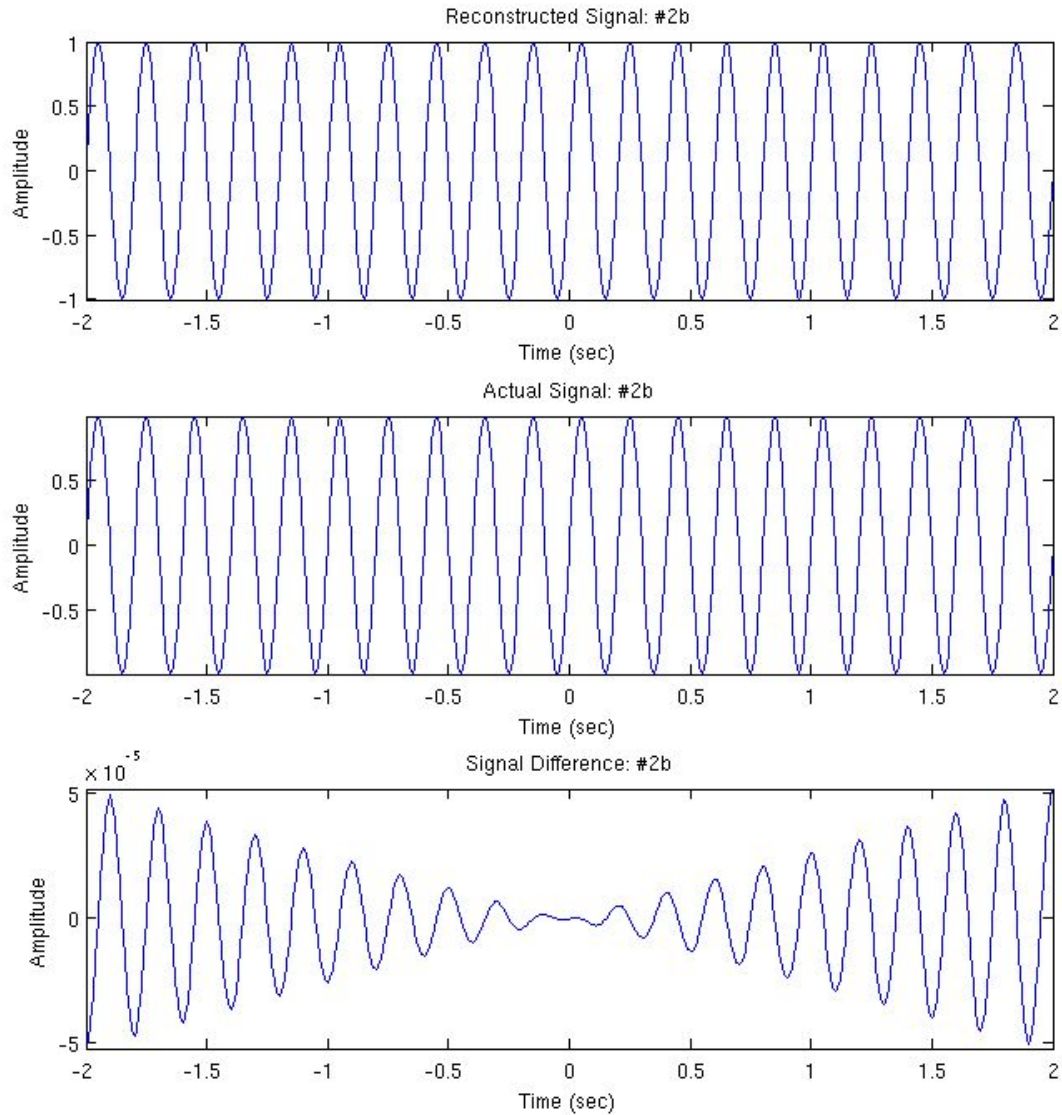
b) Plot  $x[n]$  obtained using a sampling frequency of 20 Hz



**Figure 4:** Time signal from number 2 sampled at 20 Hz



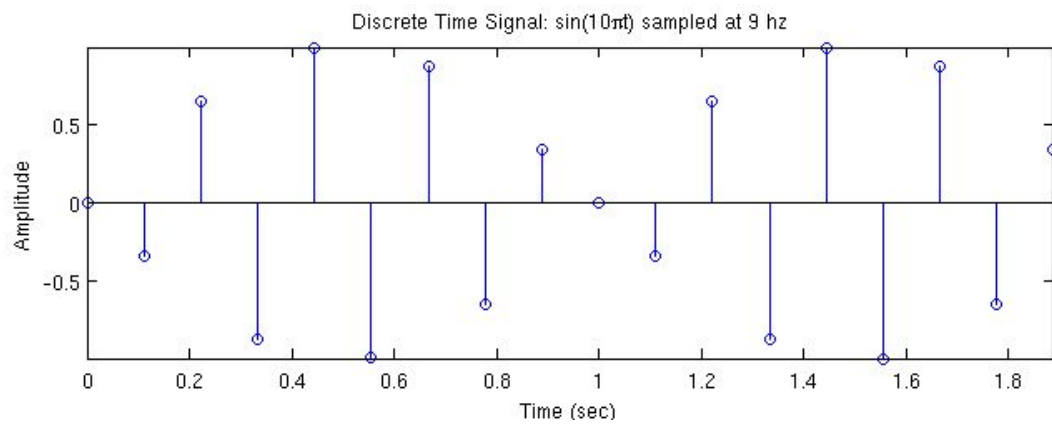
**Figure 5:** Amplitude spectrum of the discrete time signal sampled at 20 Hz



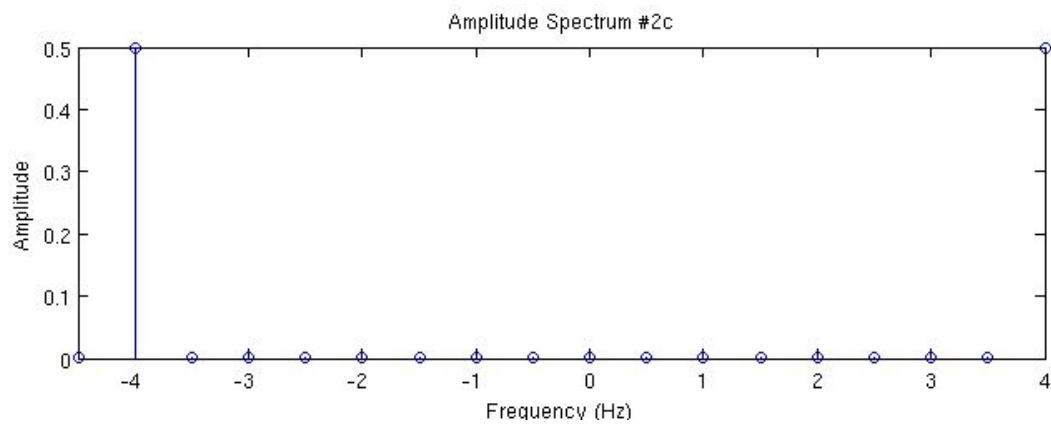
**Figure 6:** Comparison of the actual and reconstructed signal from number 2 sampled at 20 *Hz*

- The amplitude spectrum in 2b (figure 5) does match the amplitude spectrum sketch in 2a (figure 3).
- The reconstructed signal and the actual time signal do appear to be the same. Notice magnitude of difference between the two signals in figure 6 (bottom). The maximum difference in amplitude between the actual and reconstructed signal is of magnitude  $5 \times 10^{-5}$  on the interval  $-2.0 \text{ sec} \leq t \leq 2.0 \text{ sec}$ . This difference between the reconstructed and actual signal is negligible so the signals are essentially identical.

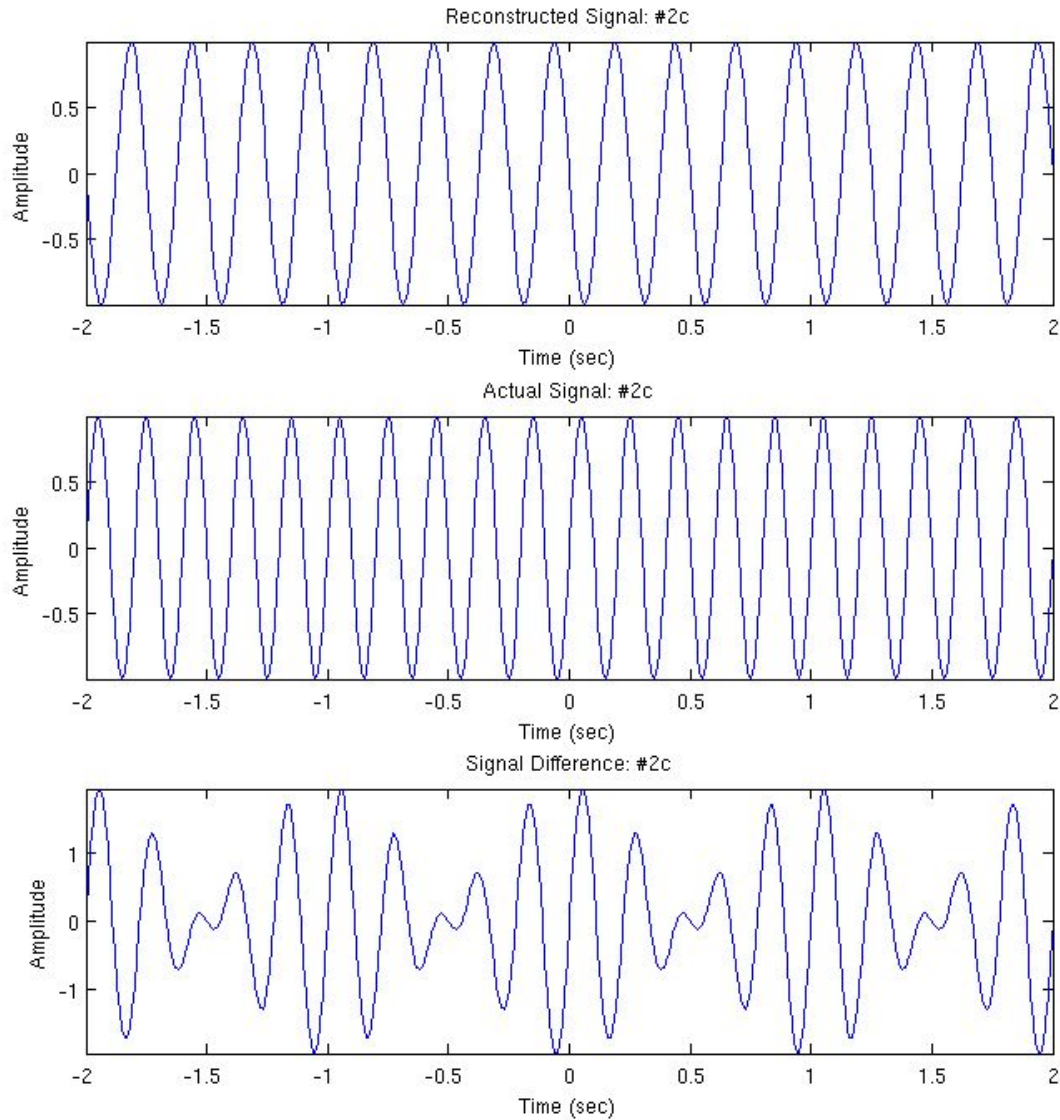
c) Plot  $x[n]$  obtained using a sampling frequency of 9 Hz



**Figure 7:** Time signal from number 2 sampled at 9 Hz



**Figure 8:** Amplitude spectrum of the discrete time signal sampled at 9 Hz



**Figure 9:** Comparison of the actual and reconstructed signal from number 2 sampled at 20 Hz

- The amplitude spectrum in 2c (figure 8) does not match the amplitude spectrum in 2a (figure 3). They do not match because the sampling frequency is below the nyquist frequency,  $f_n = 2f_{max}$  where  $f_{max} = 5 \text{ Hz}$  yields  $f_n = 10 \text{ Hz}$ . Since  $f_s < f_n$  the signal is aliased as a sinusoid of frequency 4 Hz. This satisfies the aliased frequency equation where  $f_{alias} = f_{sample} - lf_0 = (9 \text{ Hz}) - (1) * (5 \text{ Hz}) = 4 \text{ Hz}$ .
- The reconstructed time signal and actual time signal are different because the fourier transform was calculated on a data set sampled below the nyquist frequency. Note the difference between the two signals plotted in figure 9 (bottom). The difference between the two signal is of magnitude  $2 * 10^0$ , five order of magnitude greater than in 2b and twice the magnitude of the signals maximum amplitude.

3) This is number 3

a) Below is my derivation of the fourier coefficients  $c_k$

Consider the boxcar function...

$$x(t) = \left\{ 1 \text{ if } 0 \leq t < \frac{T_0}{2}, 0 \text{ if } \frac{T_0}{2} \leq t < T_0 \right\}$$

Using the analytic solution for the complex fourier coefficient  $c_k$  we find...

$$c_k = \frac{1}{T_0} \int_0^{T_0} e^{-i\frac{2\pi kt}{T_0}} dt = \frac{1}{T_0} \left( \int_0^{\frac{1}{2}T_0} e^{-i\frac{2\pi kt}{T_0}} dt + \int_{\frac{1}{2}T_0}^{T_0} e^{-i\frac{2\pi kt}{T_0}} dt \right) = \frac{1}{T_0} \int_0^{\frac{1}{2}T_0} e^{-i\frac{2\pi kt}{T_0}} dt$$

Now we solve for the DC component of our signal, the  $0^{th}$  harmonic  $c_0$  ...

$$c_0 = \frac{1}{T_0} \int_0^{\frac{1}{2}T_0} e^{-i\frac{2\pi(0)t}{T_0}} dt = \frac{1}{T_0} \int_0^{\frac{1}{2}T_0} dt = \frac{1}{T_0} \left( \frac{T_0}{2} - 0 \right) = \frac{1}{2}$$

In general the complex fourier coefficient  $c_k$  is defined as...

$$c_k = \frac{1}{T_0} \int_0^{\frac{1}{2}T_0} e^{-i\frac{2\pi kt}{T_0}} dt = \frac{1}{-2\pi ik} (e^{-i\pi k} - 1) = \frac{1}{-2\pi ik} [(-1)^k - 1]$$

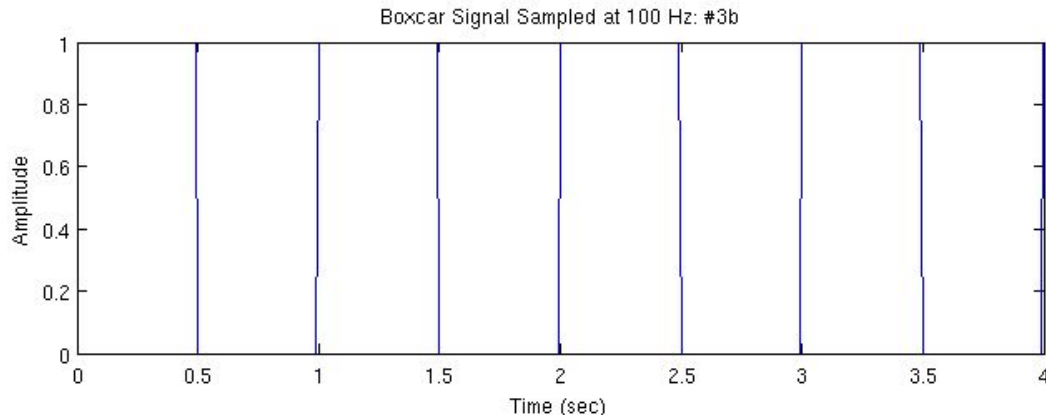
We can simplify this expression for even and odd numbers...

$$c_{even} = \frac{1}{-2\pi ik} [(-1)^{2n} - 1] = \frac{1}{-2\pi ik} [1 - 1] = 0$$
$$c_{odd} = \frac{1}{-2\pi ik} [(-1)^{2n+1} - 1] = \frac{1}{-2\pi ik} [1 * (-1)^1 - 1] = \frac{1}{-2\pi ik} [-2] = \frac{1}{\pi ik}$$

In conclusion...

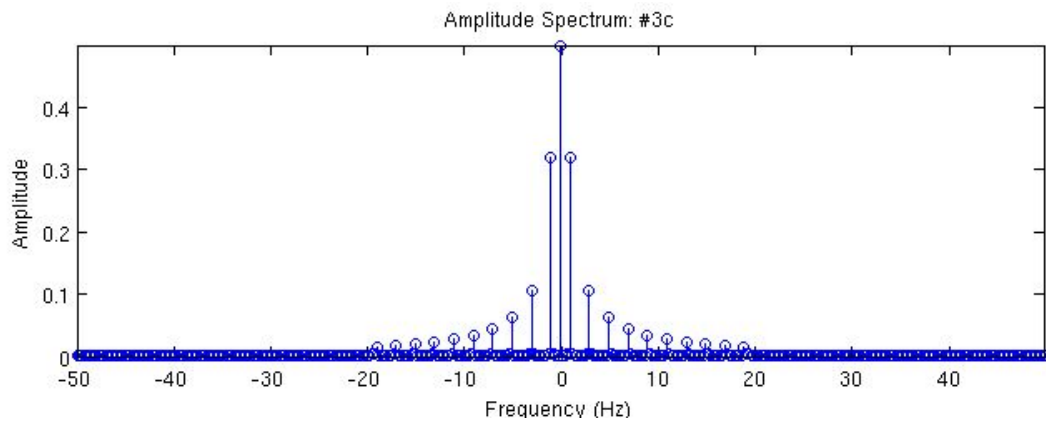
$$c_0 = \frac{1}{2}$$
$$c_{odd} = \frac{1}{\pi ik}$$
$$c_{even} = 0$$

b) Plot of  $x(t)$  on the interval  $t = [0, 4] \text{ sec}$



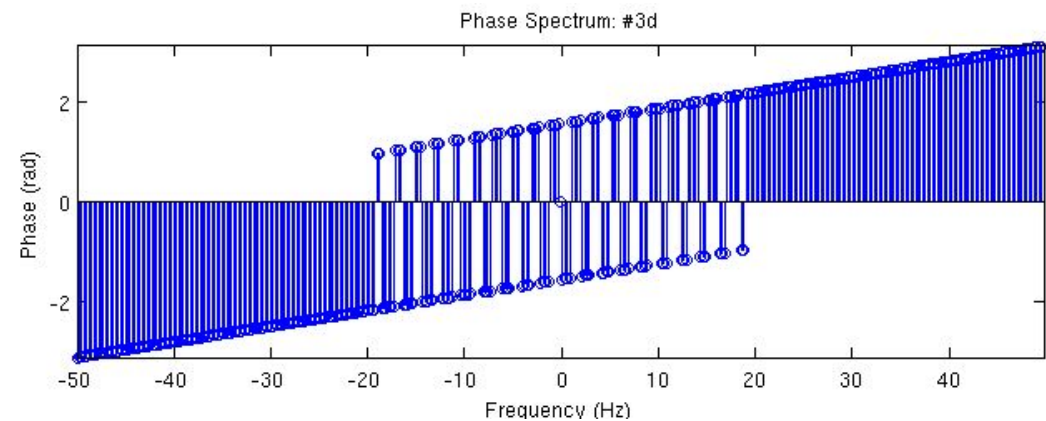
**Figure 10:** Boxcar function from 3a sampled at 100 Hz

c) Two sided amplitude spectrum for the first twenty harmonics of  $x(t)$



**Figure 11:** Amplitude spectrum of the boxcar function from 3a sampled at 100 Hz

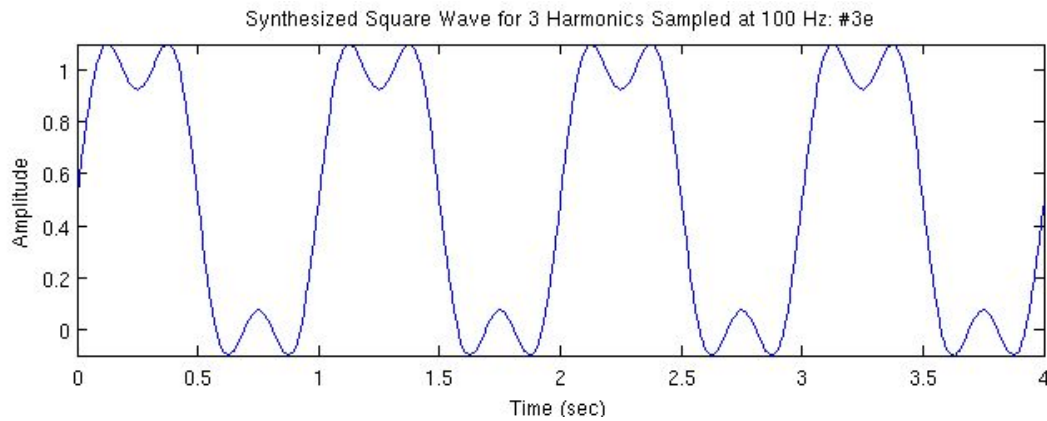
d) Two sided phase spectrum for the first twenty harmonics of  $x(t)$



**Figure 12:** Phase spectrum of the boxcar function from 3a sampled at 100 Hz

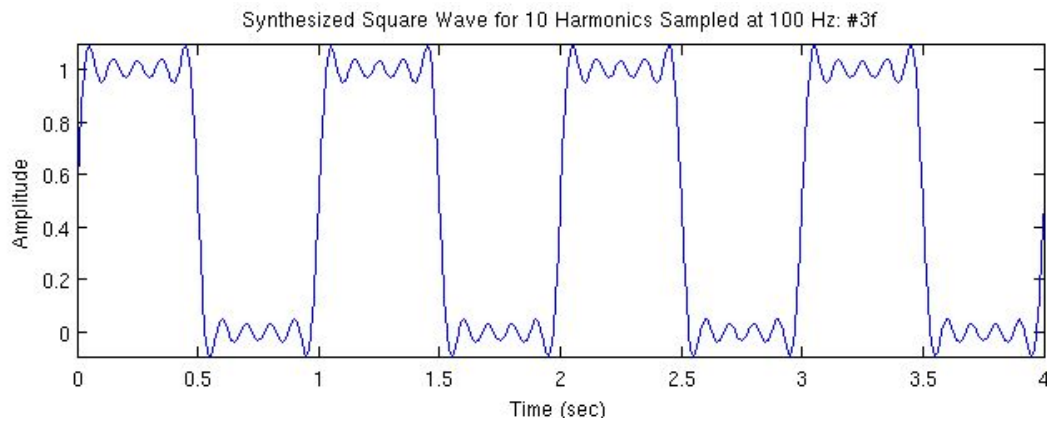


e) Synthesized time signal using the first three harmonics of  $x(t)$



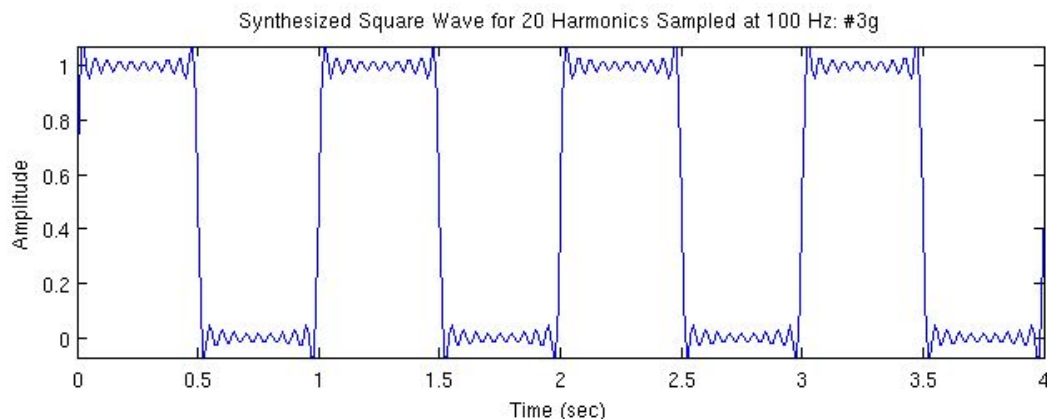
**Figure 13:** Synthesized boxcar function from 3a using the first three harmonics

f) Synthesized time signal using the first ten harmonics of  $x(t)$



**Figure 14:** Synthesized boxcar function from 3a using the first ten harmonics

g) Synthesized time signal using the first twenty harmonics of  $x(t)$



**Figure 15:** Synthesized boxcar function from 3a using the first twenty harmonics

## Appendix I: The Code

```
clear;

filename = 'data.dat';
data = tblread(filename);
time = data(:,1);
value = data(:,2);

% ----- 1a -----
%Fs = length(time) / time(length(time));
Fs = 1 / 0.020202;

Xk = fftshift(fft(value)) / length(time);
Fk = -Fs/2:Fs/length(time):Fs/2;
Fk = Fk(1:length(time));

figure('position', [0, 0, 750, 250]);
plot(time, value);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Time Signal: ',filename]);

figure('position', [0, 0, 750, 250]);
stem(Fk, abs(Xk));
axis tight;
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title(['Amplitude Spectrum: ',filename]);

DC = 0.0;
for i = 1:length(Fk)
    if Fk(i) == 0.0 || (Fk(i+1) > 0 && Fk(i) < 0)
        DC = Xk(i+1);
        break;
    end
end

% ----- 2b -----
Fs = 20;
Ts = 1.0 / Fs;

t1 = 0.0;
t2 = 2.0;
time = t1:Ts:t2;
if time(length(time)) >= 2.0
    time = time(1:(length(time)-1));
end

Xt = sin(10*3.1415926*time);
```

```

figure('position', [0, 0, 750, 250]);
stem(time, Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Discrete Time Signal: sin(10\pit) sampled at ', num2str(Fs), ' hz']);

Xk = fftshift(fft(Xt)) / length(time);
Fk = -Fs/2:Fs/length(time):Fs/2;
Fk = Fk(1:length(time));

figure('position', [0, 0, 750, 250]);
stem(Fk, abs(Xk));
axis tight;
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Amplitude Spectrum: #2b');

time = -2.0:0.01:2.0;
Xr = zeros(1,length(time));
Xt = sin(10*3.1415926*time);
for i=length(Fk)/2:length(Fk)
    Xr = Xr + Xk(i) * exp(2*pi*3.14159*Fk(i)*time);
    Xr = Xr + conj(Xk(i)) * exp(-2*pi*3.14159*Fk(i)*time);
end

figure('position', [0, 0, 750, 750]);
subplot(3,1,1);
plot(time, Xr);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Reconstructed Signal: #2b');
subplot(3,1,2);
plot(time, Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Actual Signal: #2b');
subplot(3,1,3);
plot(time, Xt-Xr);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Signal Difference: #2b');

% ----- 2c -----
Fs = 9;
Ts = 1.0 / Fs;

```

```

t1 = 0.0;
t2 = 2.0;
time = t1:Ts:t2;
if time(length(time)) >= 2.0
    time = time(1:(length(time)-1));
end

Xt = sin(10*3.1415926*time);

figure('position', [0, 0, 750, 250]);
stem(time, Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Discrete Time Signal: sin(10\pi t) sampled at ', num2str(Fs), ' Hz']);

Xk = fftshift(fft(Xt)) / length(time);
Fk = -Fs/2:Fs/length(time):Fs/2;
Fk = Fk(1:length(time));

figure('position', [0, 0, 750, 250]);
stem(Fk, abs(Xk));
axis tight;
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Amplitude Spectrum #2c');

time = -2.0:0.01:2.0;
Xr = zeros(1,length(time));
Xt = sin(10*3.1415926*time);

for i=length(Fk)/2:length(Fk)
    Xr = Xr + Xk(i) * exp(2*pi*Fk(i)*time);
    Xr = Xr + conj(Xk(i)) * exp(-2*pi*Fk(i)*time);
end

figure('position', [0, 0, 750, 750]);
subplot(3,1,1);
plot(time, Xr);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Reconstructed Signal: #2c');
subplot(3,1,2);
plot(time, Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Actual Signal: #2c');

```

```

subplot(3,1,3);
plot(time, Xt-Xr);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title('Signal Difference: #2c');

% ----- 3b -----
Fs = 100;
period = 1.0;
time = 0.0:1/Fs:4.0;

figure('position', [0, 0, 750, 250]);
plot(time, mod(time, period) < period*0.5);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Boxcar Signal Sampled at ', num2str(Fs), ' Hz: #3b']);

% ----- 3c + 3d -----
harmonic = 20;
Fs = 100;
period = 1.0;
time = 0.0:1/Fs:4.0;

Xt = zeros(1,length(time));

for k=-harmonic:harmonic
    if mod(k,2)
        Xt = Xt + ((1i*3.14159*k).^(-1)*exp((2i*3.14159*k*time)/period));
    end
end

Xt = Xt + 0.5;

Xk = fftshift(fft(Xt)) / length(time);
Fk = -Fs/2:Fs/length(time):Fs/2;
Fk = Fk(1:length(time));

figure('position', [0, 0, 750, 250]);
stem(Fk, abs(Xk));
axis tight;
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Amplitude Spectrum: #3c');

figure('position', [0, 0, 750, 250]);
stem(Fk, angle(Xk));
axis tight;
xlabel('Frequency (Hz)');

```

```

ylabel('Phase (rad)');
title('Phase Spectrum: #3d');

% ----- 3e/f/g -----
harmonic = 100;
Fs = 100;
period = 1.0;
time = 0.0:1/Fs:4.0;

harmonic_matrix = zeros(2*harmonic+1, length(time));
for k=-harmonic:harmonic
    row = k + 1 + harmonic;
    if mod(k,2)
        harmonic_matrix(row,:) =
(1i*3.14159*k).^(-1)*exp((2i*3.14159*k*time)/period);
    elseif k == 0
        harmonic_matrix(row,:) = harmonic_matrix(row) + 0.5;
    else
        harmonic_matrix(row,:) = 0.0;
    end
end

% ----- 3e -----
h = 3;
Xt = zeros(1,length(time));
for k=-h:h
    row = k + 1 + harmonic;
    Xt = Xt + harmonic_matrix(row,:);
end
figure('position', [0, 0, 750, 250]);
plot(time,Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Synthesized Square Wave for ', num2str(h),' Harmonics Sampled at ',
num2str(Fs), ' Hz: #3e']);

% ----- 3f -----
h = 10;
Xt = zeros(1,length(time));
for k=-h:h
    row = k + 1 + harmonic;
    Xt = Xt + harmonic_matrix(row,:);
end
figure('position', [0, 0, 750, 250]);
plot(time,Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');

```

```

title(['Synthesized Square Wave for ', num2str(h),' Harmonics Sampled at ',
num2str(Fs), ' Hz: #3f']);

% ----- 3g -----
h = 20;
Xt = zeros(1,length(time));
for k=-h:h
    row = k + 1 + harmonic;
    Xt = Xt + harmonic_matrix(row,:);
end
figure('position', [0, 0, 750, 250]);
plot(time,Xt);
axis tight;
xlabel('Time (sec)');
ylabel('Amplitude');
title(['Synthesized Square Wave for ', num2str(h),' Harmonics Sampled at ',
num2str(Fs), ' Hz: #3g']);

```