Constructor University Bremen

CO-520-B

Signals & Signals Lab

Fall Semester 2023

Lab Experiment 3 – Fourier Series and Fourier Transform

Experiment conducted by: Abhigyan Deep Barnwal, Sumit Giri

Place of execution: Room 54, Research I, Jacobs University, 28759 Bremen

Date of execution: 11th October 2023

Author of the Report: Abhigyan Deep Barnwal

IRC Mailbox Number: NB-124

Introduction

In this lab, we gained a greater knowledge of the Fourier Transform by examining the various signals in terms of their Fourier coefficients. Both the frequency domain and the time domain can be used to represent a signal. The process of breaking down a signal into sinusoids is known as Fourier analysis, and the frequency domain representation of the data is referred to as the signal spectrum.

By layering sinusoidal waves on top of one another, any time-varying signal can be created. A signal can be converted from the time domain to the frequency domain using the Fourier Transform. A discrete Fourier transform is needed to digitize arbitrary signals; otherwise, continuous can be applied to a class of signals.

We could also use the inverse Fourier Transform to convert a signal from the frequency domain back to the time domain.

Prelab - Fourier Series and Fourier Transform

Problem 1 - Decibels

1. Given $x(t) = 5cos(2\pi 1000t)$
   a. What is the signal amplitude and the V_{pp} voltage?
   $$x(t) = 5cos(2\pi 1000t)$$
   $$-5 \le cos(\theta) \le 5$$
   $$V_{pp} = 10V$$
   b. What is the V_{RMS} value of the provided signal?
   $$V_{RMS} = \sqrt{\frac{1}{T}\int_{t_0}^{t_0+T} V^2(t)dt}$$
   $$V_{RMS} = \sqrt{\frac{1}{10^{-3}}\int_{0}^{10^{-3}} 25cos^2(2\pi 1000t)dt}$$
   $$V_{RMS} = 3.53553$$
   c. What is the amplitude of the spectral peak in dBV_{RMS}?
   $$dBV_{RMS} = 20log(V_{RMS}) = 10.9691$$
2. For a square wave of $1V_{pp}$ the voltage changes between -0.5V and 0.5V

a. What is the signal amplitude in V_{RMS}?

$$V_{RMS} = \sqrt{\frac{1}{T}\int_{t_0}^{t_0+T} V^2(t)dt}$$

$$V_{RMS} = \sqrt{\frac{1}{1}\int_{-0.5}^{0.5} 1dt} = 1V$$

b. What is the amplitude in dBV_{RMS}?

$$dBV_{RMS} = 20log(V_{RMS}) = 0$$

Problem 2 - Determination of Fourier series coefficients

1. Determine the Fourier Series coefficients up to the 5th harmonic of the function

   $f(t) = 4t^2 \qquad -0.5 < t < 0.5$

The given function is even, so $b_n = 0$.

$$a_0 = \frac{1}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} f(t)dt$$

$$a_0 = \frac{1}{1}\int_{-0.5}^{0.5} 4t^2dt$$

$$a_0 = \frac{1}{3}$$

$$a_n = \frac{2}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} f(t)cos(n\omega_0 t)dt$$

$$a_n = \frac{2}{1}\int_{-0.5}^{0.5} 4t^2cos(n\omega_0 t)dt$$

$$a_n = \frac{8}{1}[\frac{t^2sin(n\omega_0 t)}{n\omega_0} + \int_{-0.5}^{0.5} 2t\frac{sin(n\omega_0 t)}{n\omega_0}dt]_{-0.5}^{0.5}$$

$$a_n = 8[\frac{t^2sin(n\omega_0 t)}{n\omega_0} + \frac{2}{n\omega_0}\int_{-0.5}^{0.5} tsin(n\omega_0 t)dt]_{-0.5}^{0.5}$$

$$a_n = 8[\frac{t^2 sin(n\omega_0 t)}{n\omega_0} + \frac{2}{n\omega_0}(\frac{-tcos(n\omega_0 t)}{n\omega_0} + \frac{-sin(n\omega_0 t)}{n^2\omega_0^{\ 2}})]_{-0.5}^{0.5}$$

$$a_n = 8[\frac{t^2 sin(n\omega_0 t)}{n\omega_0} + \frac{-2tcos(n\omega_0 t)}{n^2\omega_0^{\ 2}} + \frac{-2sin(n\omega_0 t)}{n^3\omega_0^{\ 3}}]_{-0.5}^{0.5}$$

$$a_n = 8[\frac{t^2 sin(n2\pi t)(n^2 4\pi^2) - 2tcos(n2\pi t)(n2\pi) - 2sin(n2\pi t)}{n^3 8\pi^3}]_{-0.5}^{0.5}$$

$$a_n = 8[\frac{0.25sin(n\pi)(n^2 4\pi^2) - cos(n\pi)(n2\pi) - 2sin(n\pi)}{n^3 8\pi^3} - \frac{0.25sin(-n\pi)(n^2 4\pi^2) + cos(-n\pi)(n2\pi) - 2sin(-n\pi)}{n^3 8\pi^3}]$$

$$a_n = \frac{-cos(n\pi)(n2\pi) - cos(-n\pi)(n2\pi)}{n^3\pi^3}$$

$$a_n = \frac{-2cos(n\pi) - 2cos(-n\pi)}{n^2\pi^2}$$

$$a_n = \frac{-4cos(n\pi)}{n^2\pi^2}$$

$$a_n = \frac{4(-1)^{n+1}}{n^2\pi^2}$$

$$f(t) = \frac{1}{3} + \frac{4}{\pi^2}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n^2}cos(n\omega_0 t)$$

$$n = 1: f(t) = \frac{1}{3} + \frac{4}{\pi^2}(cos(\omega_0 t))$$

$$n = 2: f(t) = \frac{1}{3} + \frac{4}{\pi^2}(cos(\omega_0 t) - \frac{1}{4}cos(2\omega_0 t))$$

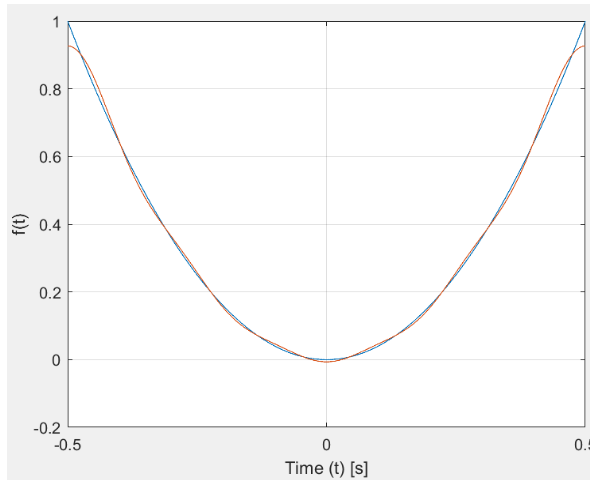
$$n = 3: f(t) = \frac{1}{3} + \frac{4}{\pi^2}(cos(\omega_0 t) - \frac{1}{4}cos(2\omega_0 t) + \frac{1}{9}cos(3\omega_0 t))$$

$$n = 4: f(t) = \frac{1}{3} + \frac{4}{\pi^2}(cos(\omega_0 t) - \frac{1}{4}cos(2\omega_0 t) + \frac{1}{9}cos(3\omega_0 t) - \frac{1}{16}cos(4\omega_0 t))$$

$$n = 5: f(t) = \frac{1}{3} + \frac{4}{\pi^2}(cos(\omega_0 t) - \frac{1}{4}cos(2\omega_0 t) + \frac{1}{9}cos(3\omega_0 t) - \frac{1}{16}cos(4\omega_0 t) + \frac{1}{25}cos(5\omega_0 t))$$

2. Use MatLab to plot the original function and the inverse Fourier transform. Put both graphs into the same diagram.

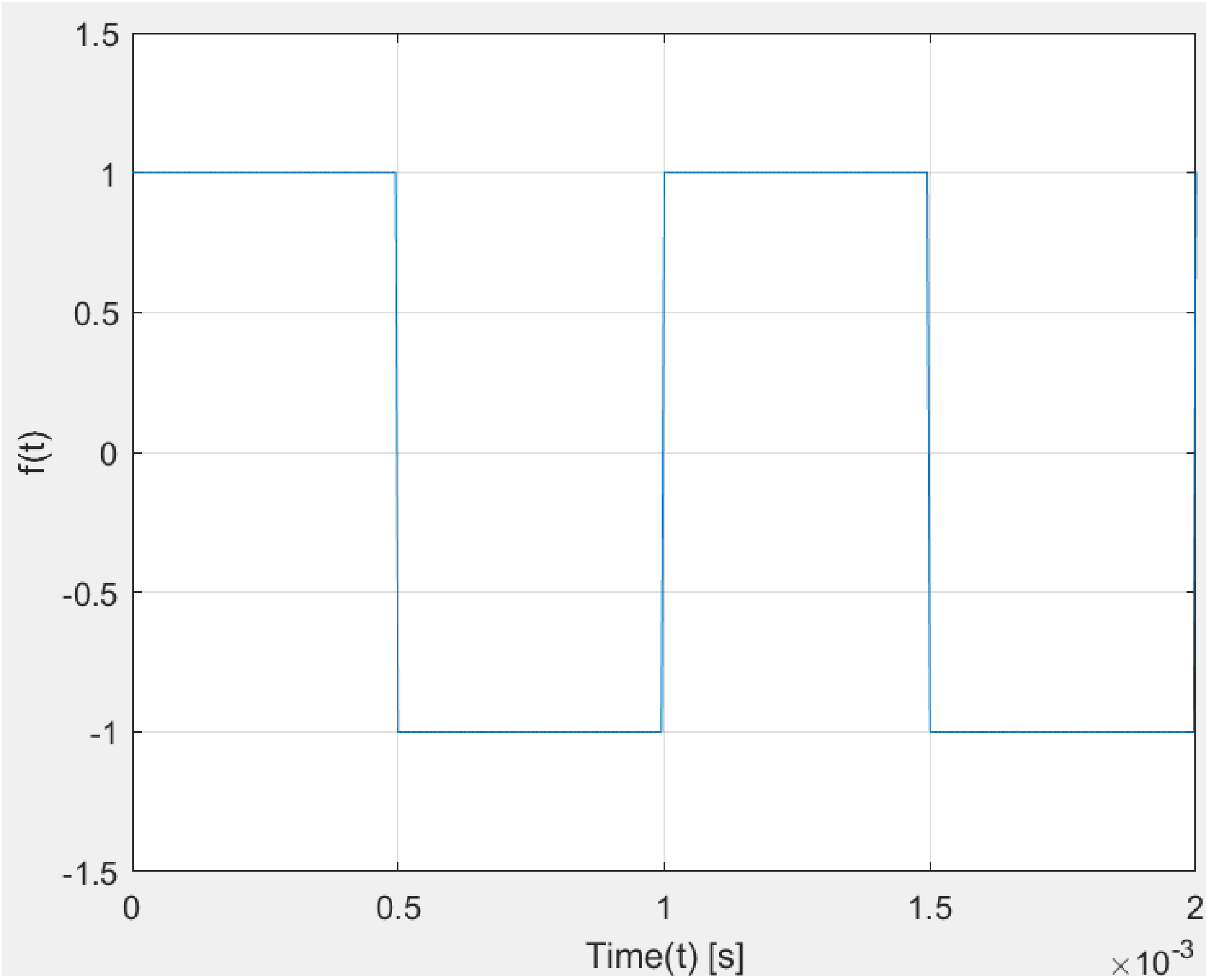
```
t=(-0.5):0.0001:0.5; %given function
y=4*t.*t;

f=1/3-(4/pi^2)*(cos(2*pi*t)-(1/4)*cos(4*pi*t)+(1/9)*cos
%fourier series approximation
figure(1);
plot(t,y);
xlim([-0.5,0.5]);
grid on;
hold on;
plot(t,f);
xlabel("Time (t) [s]")
ylabel("f(t)");
```



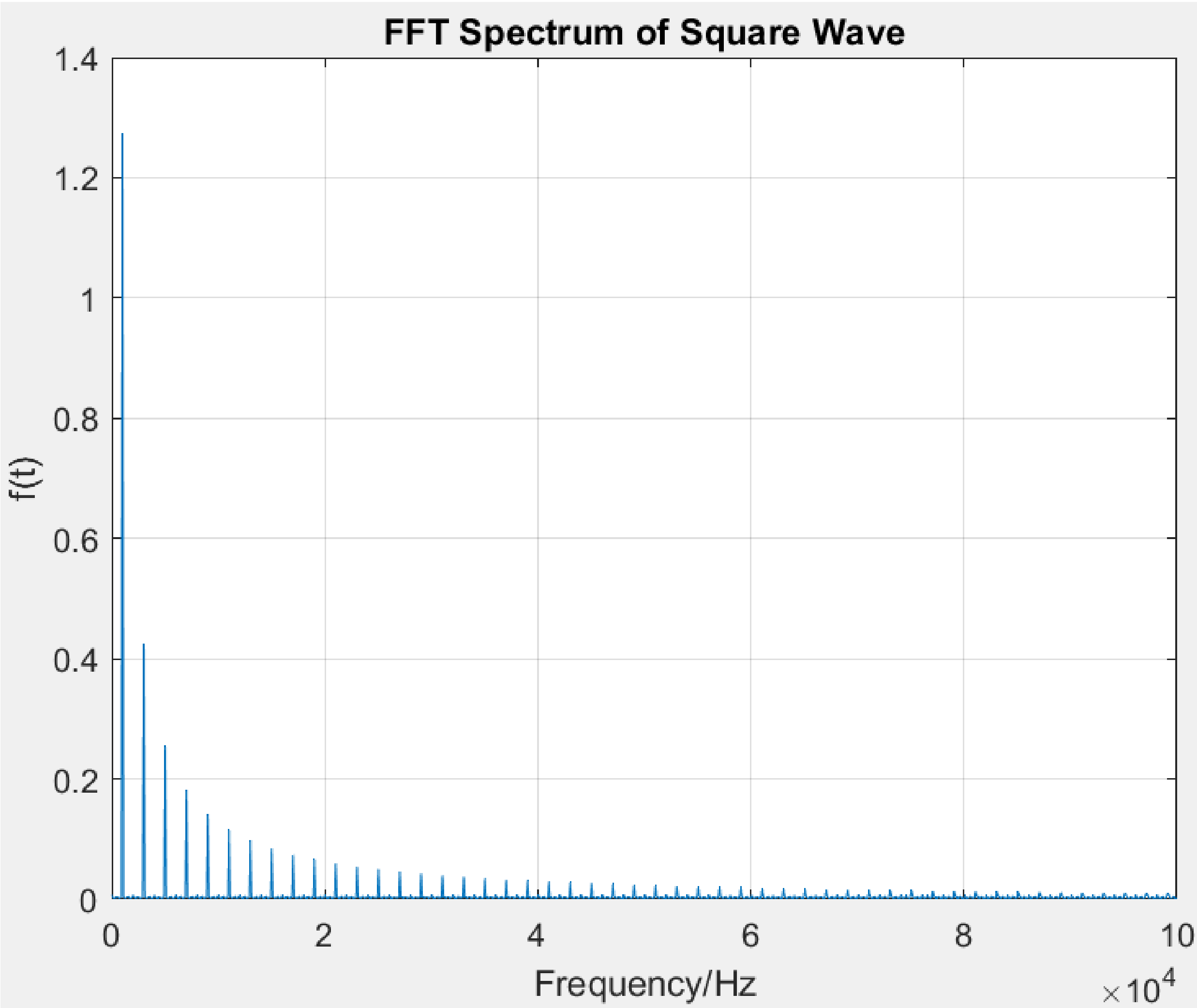
Problem 3 - FFT of a Square/Rectangular Wave

Write a MATLAB script:

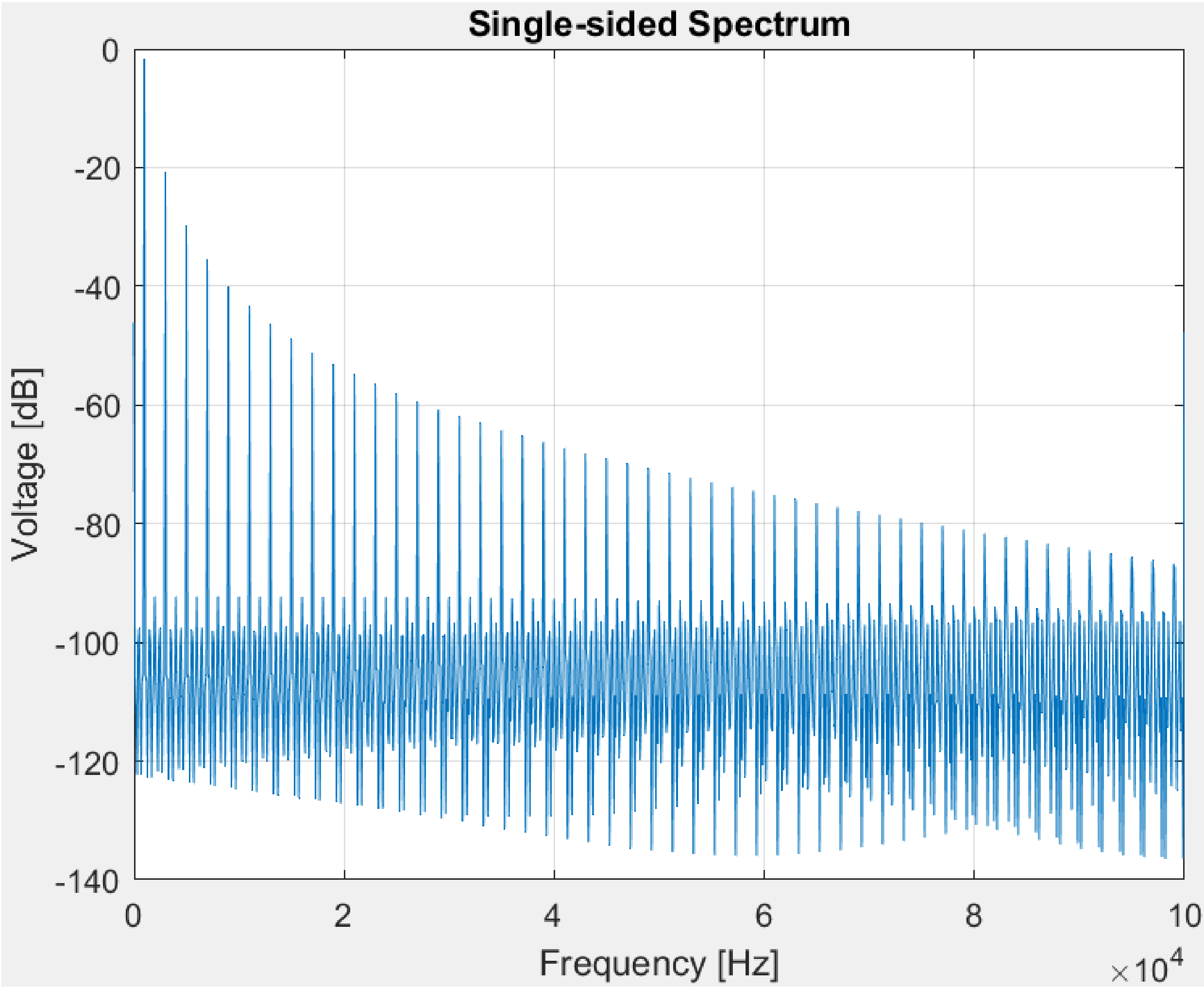
Generate a square wave of 1 ms period, 2 Vpp amplitude, no offset, and duty cycle 50% (hint : use 'square' function). Use 200 kHz as the sampling frequency for the problem. Plot the square wave in time domain. Obtain the FFT spectrum using Matlab FFT function. Make the FFT length to be the length of the square wave data vector. Plot the single-sided amplitude spectrum in dBVrms. Plot the spectrum including only the first four harmonics in dBVrms. Hint: Use the Matlab command 'xlim'. Repeat the previous steps using 20% and 33% duty cycles, respectively. Keep period and amplitude constant. Discuss the changes for smaller pulse width. Use Eq. (6.11) to prove your statement. Hint: Use the subplot command to plot the spectrum magnitude for the three cases 50%, 33% and 20% duty cycle to ease the comparison.



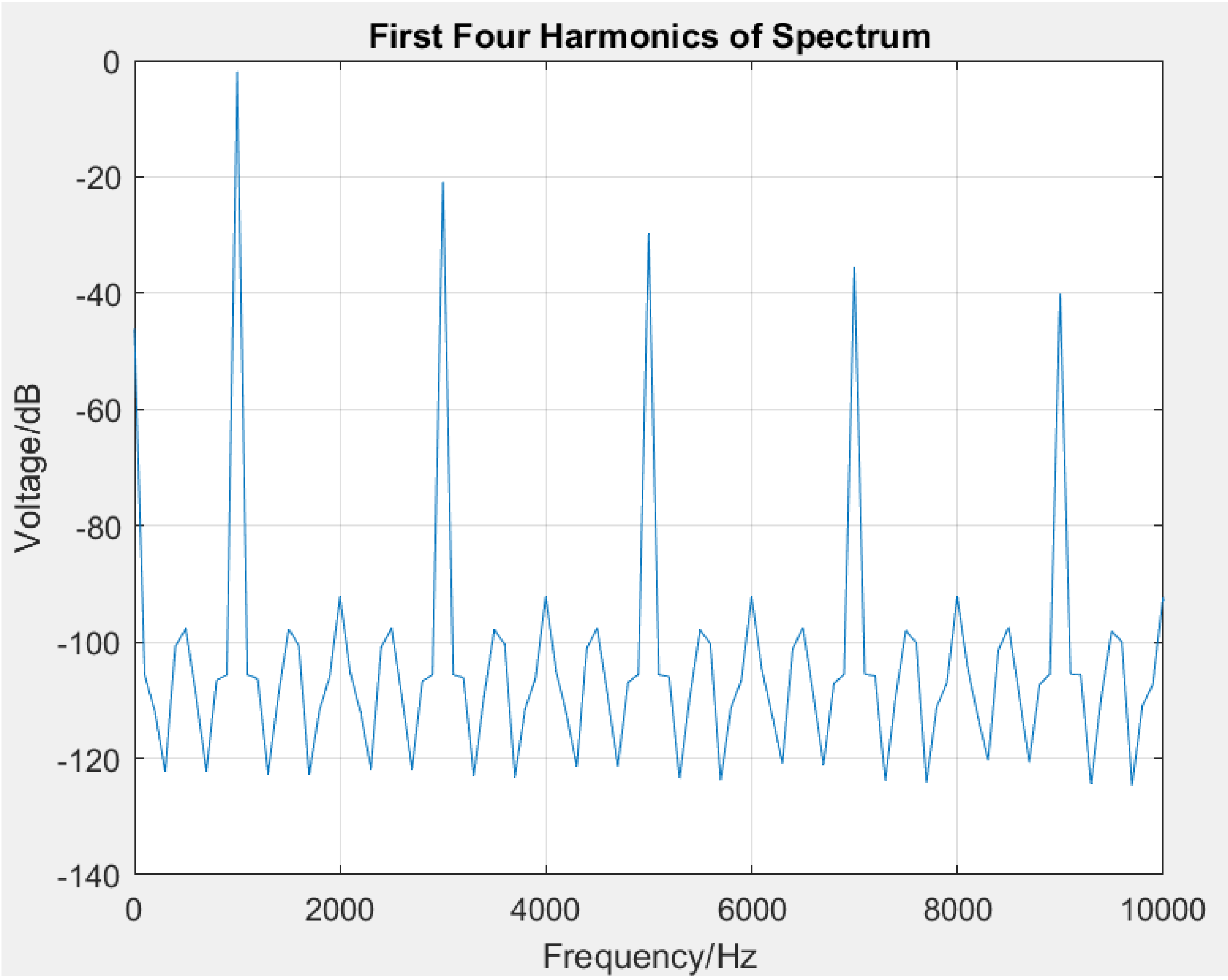

```
%Square wave
Fs = 200000;
t = 0:1/Fs:0.01;
sq = square((2*pi*t)/(10^-3), 50);
figure(1);
plot(t,sq);
xlim([0,0.002])
ylim([-1.5,1.5])
xlabel("Time(t) [s]")
ylabel("f(t)")
grid on;
```



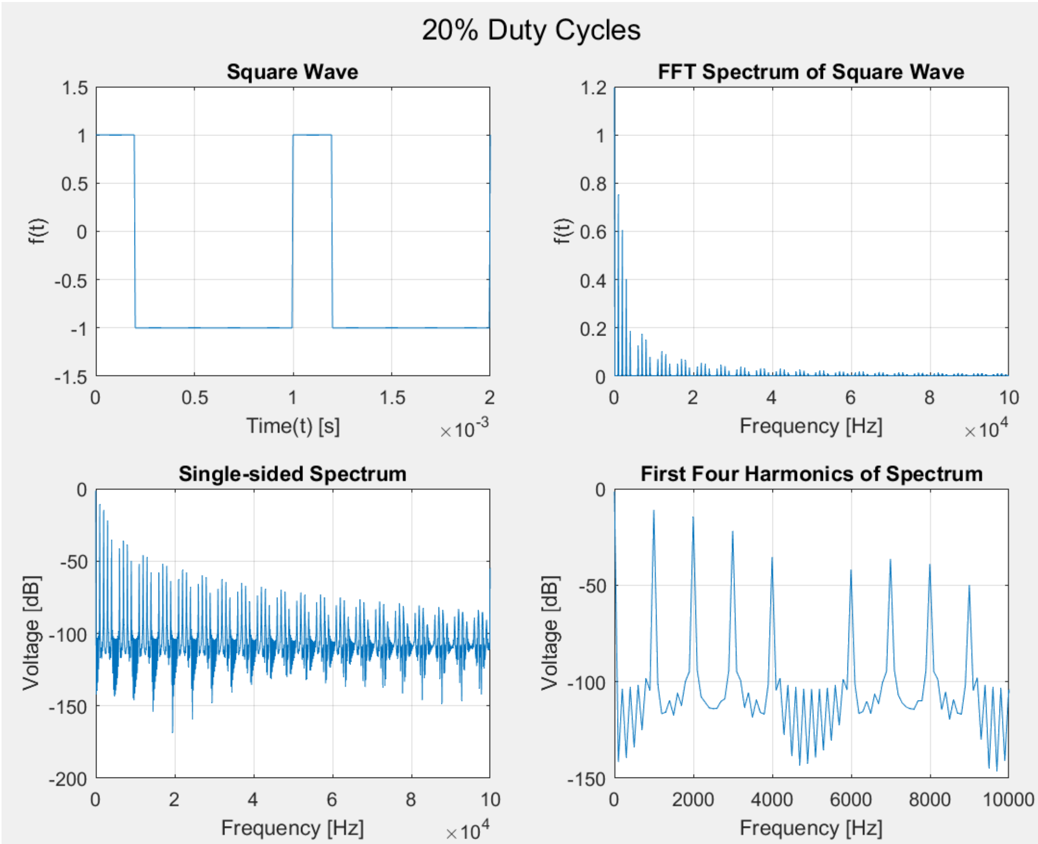

```
%FFT
L = length(sq);
F_nyq = Fs/2;
sq_fft = fft(sq);
sq_fft = 2*abs(sq_fft)/L;
sq_fft = sq_fft(1:L/2);
f = linspace(0, F_nyq, L/2);
figure(2)
plot(f, sq_fft);
xlabel('Frequency/Hz');
ylabel('f(t)');
title("FFT Spectrum of Square Wave");
grid on;
```




```
%Single Sided Spectrum
f = Fs*(0:(length(sq)/2))/length(sq);
SSSpec = abs(2*fft(sq)/length(sq));
SSSpec = db(SSSpec/sqrt(2));
SSSpec = SSSpec(1:length(sq)/2+1);
SSSpec(2:end-1) = 2*SSSpec(2:end-1);
figure (3);
plot(f, SSSpec);
xlabel('Frequency [Hz]');
ylabel('Voltage [dB]');
title("Single-sided Spectrum");
grid on;
```

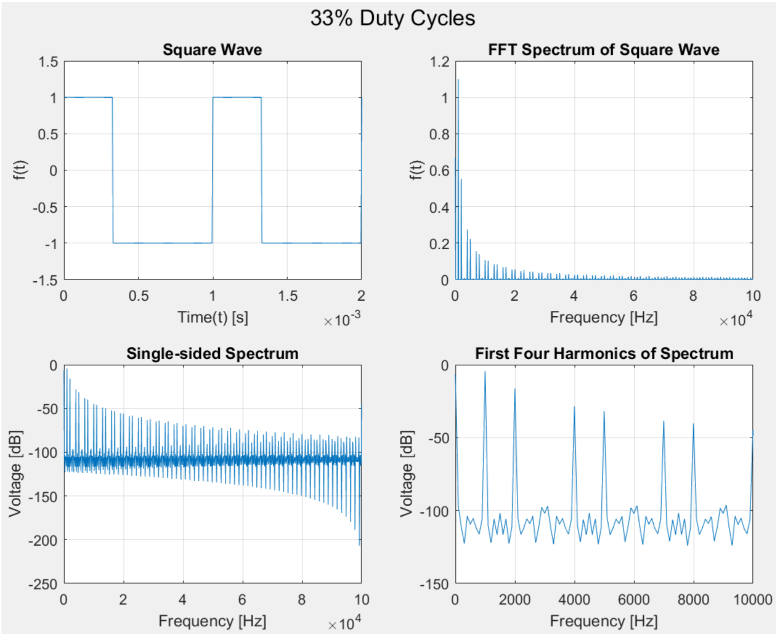



```
%Spectrum with only the first four harmonics
figure (4);
plot(f, SSSpec);
xlabel('Frequency/Hz'); %labeling Axes
ylabel('Voltage/dB');
title("First Four Harmonics of Spectrum");
xlim([0,10000]);
grid on;
```




```
%Square wave
Fs = 200000;
t = 0:1/Fs:0.01;
sq = square((2*pi*t)/(10^-3), 20);
subplot(2,2,1);
plot(t,sq);
xlim([0,0.002]);
ylim([-1.5,1.5]);
xlabel("Time(t)  [s]");
ylabel("f(t)");
grid on;
title("Square Wave")
%FFT
L = length(sq);
F_nyq = Fs/2;
sq_fft = fft(sq);
sq_fft = 2*abs(sq_fft)/L;
sq_fft = sq_fft(1:L/2);
f = linspace(0, F_nyq, L/2);
subplot(2,2,2);
plot(f, sq_fft);
xlabel('Frequency [Hz]');
ylabel('f(t)');
title("FFT Spectrum of Square Wave");
grid on;
%Single Sided Spectrum
f = Fs*(0:(length(sq)/2))/length(sq);
```

```
%Single Sided Spectrum
f = Fs*(0:(length(sq)/2))/length(sq);
SSSpec = abs(2*fft(sq)/length(sq));
SSSpec = db(SSSpec/sqrt(2));
SSSpec = SSSpec(1:length(sq)/2+1);
SSSpec(2:end-1) = 2*SSSpec(2:end-1);
subplot(2,2,3);
plot(f, SSSpec);
xlabel('Frequency [Hz]');
ylabel('Voltage [dB]');
title("Single-sided Spectrum");
grid on;
%Spectrum with only the first four harmonics
subplot(2,2,4);
plot(f, SSSpec);
xlabel('Frequency [Hz]'); %labeling Axes
ylabel('Voltage [dB]');
title("First Four Harmonics of Spectrum");
xlim([0,10000]);
grid on;
```



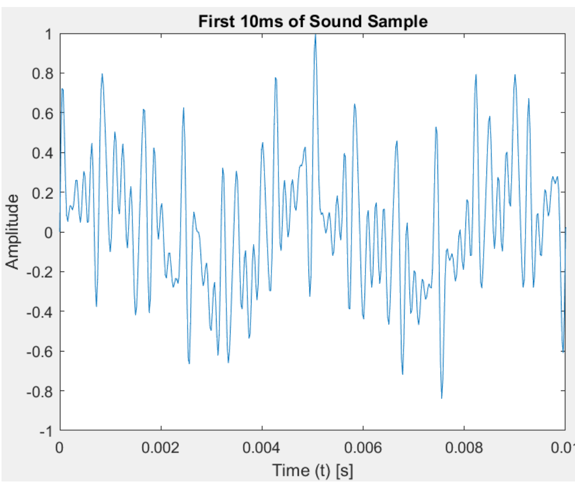

```
%Square wave
Fs = 200000;
t = 0:1/Fs:0.01;
sq = square((2*pi*t)/(10^-3), 33);
sgtitle("33% Duty Cycles");
subplot(2,2,1);
plot(t,sq);
xlim([0,0.002]);
ylim([-1.5,1.5]);
xlabel("Time(t) [s]");
ylabel("f(t)");
grid on;
title("Square Wave")
%FFT
L = length(sq);
F_nyq = Fs/2;
sq_fft = fft(sq);
sq_fft = 2*abs(sq_fft)/L;
sq_fft = sq_fft(1:L/2);
f = linspace(0, F_nyq, L/2);
subplot(2,2,2);
plot(f, sq_fft);
xlabel('Frequency [Hz]');
ylabel('f(t)');
title("FFT Spectrum of Square Wave");
grid on;
%Single Sided Spectrum
```

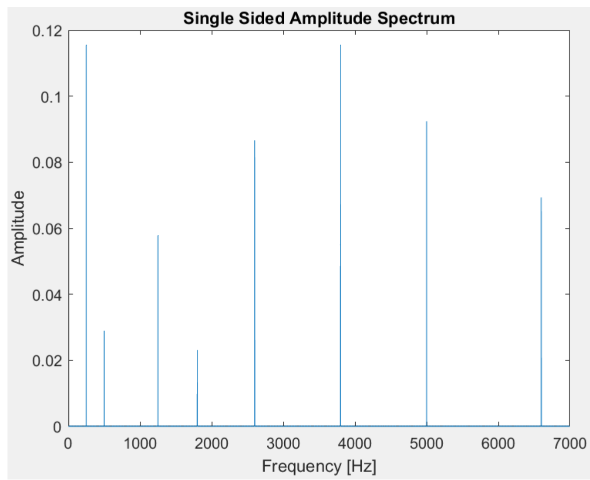
```
%Single Sided Spectrum
f = Fs*(0:(length(sq)/2))/length(sq);
SSSpec = abs(2*fft(sq)/length(sq));
SSSpec = db(SSSpec/sqrt(2));
SSSpec = SSSpec(1:length(sq)/2+1);
SSSpec(2:end-1) = 2*SSSpec(2:end-1);
subplot(2,2,3);
plot(f, SSSpec);
xlabel('Frequency [Hz]');
ylabel('Voltage [dB]');
title("Single-sided Spectrum");
grid on;
%Spectrum with only the first four harmonics
subplot(2,2,4);
plot(f, SSSpec);
xlabel('Frequency [Hz]'); %labeling Axes
ylabel('Voltage [dB]');
title("First Four Harmonics of Spectrum");
xlim([0,10000]);
grid on;
```

Analyzing the equation, we observe a direct relationship between the pulse width (T1) of a function and the magnitude of its spectrum. Specifically, a lower pulse width results in a higher magnitude, while a higher pulse width corresponds to a lower magnitude. This phenomenon is evident in our subplots, where a 20% duty cycle exhibits the highest spectrum magnitude, whereas a 50% duty cycle showcases the lowest. In the given equation, T represents the period of the signal, and T1 denotes the width of periodic impulses. As the duty cycle decreases, the magnitude spectra of the square wave approach zero for multiples of w0. Notably, reducing the pulse width T1 leads to an increased magnitude of the spectrum. In our observations, a 20% duty cycle yielded the highest magnitude spectrum, while a 50% duty cycle resulted in the lowest magnitude. Furthermore, the decrease in duty cycle causes parts of the input signal to come closer together, introducing additional Fourier series coefficients. This closer alignment contributes to a larger amplitude in the spectrum, emphasizing the intricate relationship between duty cycle, pulse width, and the resulting spectral characteristics of the signal.

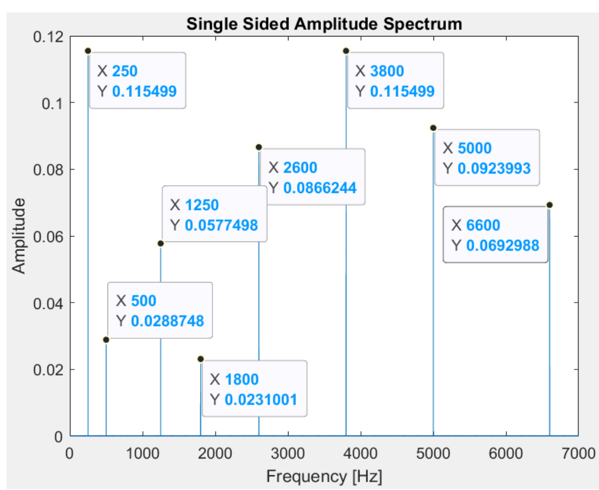
Problem 4 - FFT of a Sound Sample

The Matlab command '[y,Fs] = audioread(filename)' reads a wave file specified by the string 'filename', returning the sampled data in 'y' and the sample rate in 'Fs' in Hertz. Download the 'Soundfile for Prelab question 'FFT of a sound sample" from the course webpage. Using Matlab, read the sound file and plot the first 10 ms of the signal. Use the Matlab FFT function to compute the spectrum and plot the singlesided amplitude spectrum in dBVrms. What are the tones forming this signal?






```
%Plotting the first 10ms
[A,Fs] = audioread("s_samp.wav");
t=0:1/Fs:(numel(A)-1)/Fs;
plot(t(1:450),A(1:450));
xlabel("Time (t) [s]");
xlim([0,0.01]);
ylabel("Amplitude");
title("First 10ms of Sound Sample");

%Plotting the Single sided spectrum
L=numel(A);
A_fft=fft(A)/L;
A_fft2=abs(A_fft(1:floor((L+1)/2)));
Domain=(0:(L-1)/2);
fDomain=Fs*Domain/L;
plot(fDomain,A_fft2);
xlim([0,4000]);
xlabel("Frequency [Hz]");
ylabel("Amplitude");
title("Single Sided Amplitude Spectrum");
```



The tones forming the signal are 250Hz, 500Hz, 1250Hz, 1800Hz, 2600Hz, 3800Hz, 5000Hz and 6600Hz.

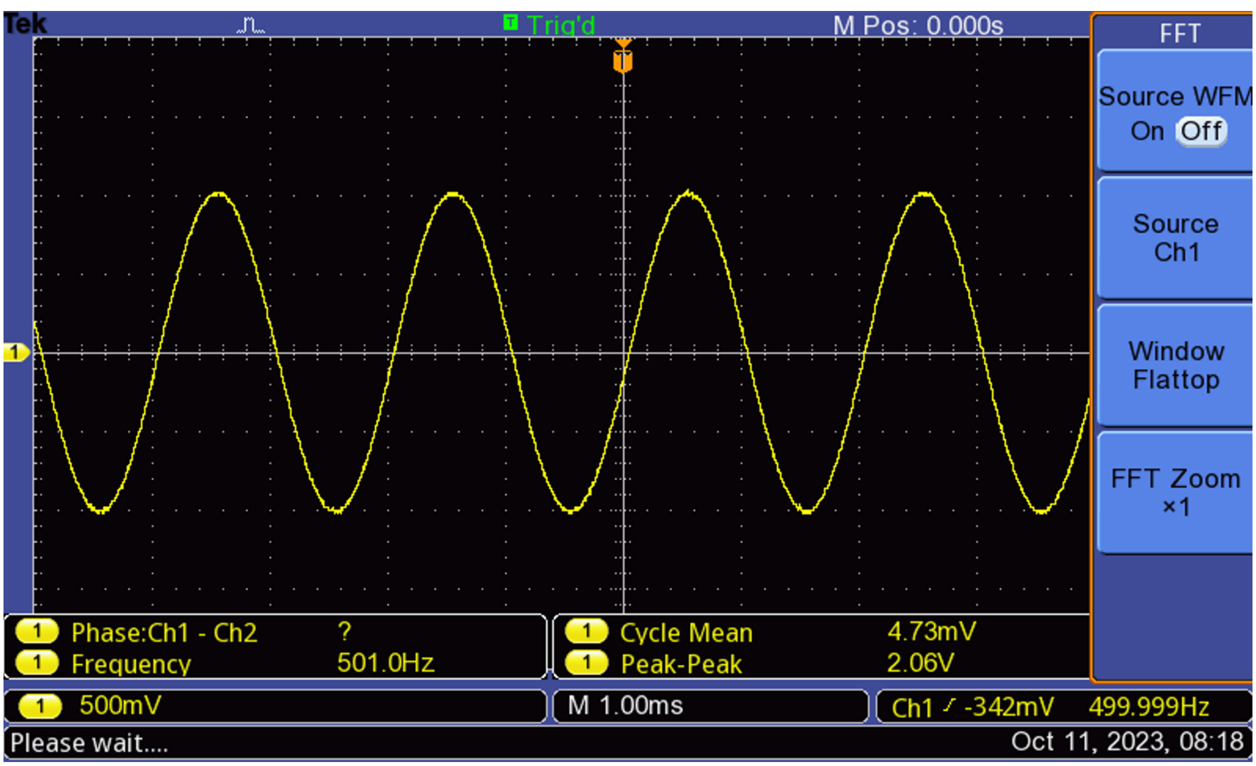
Experimental Set-up and Results

Part 1 - FFT of a single-tone sinusoidal wave

Experimental Setup and Procedure

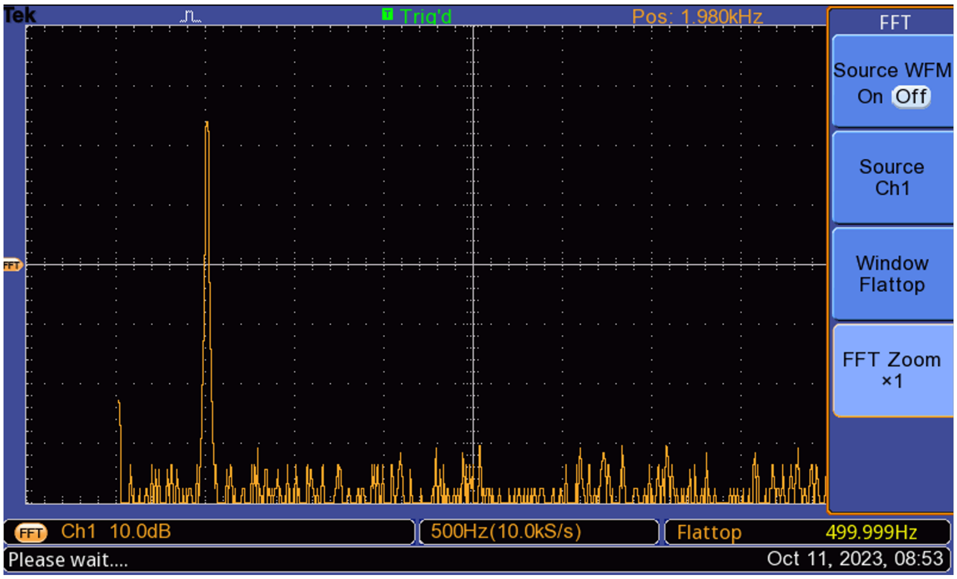
1. Use the function generator to generate a sinusoidal wave having 500 Hz frequency, 2 Vpp amplitude and no offset. Use the measure function to verify all properties. Take a hard copy in time domain.
2. Obtain the FFT spectrum using the oscilloscope FFT function. Use the cursor to measure the properties. Take hard copies of the complete spectra and the zoomed spectra peak.
3. Generate a sinusoidal wave having 0 dB spectrum peak, 2 KHz frequency, without a dc offset. What is the amplitude value? Use the measure function and the cursors. Take hard copies of time and frequency domain.

Results

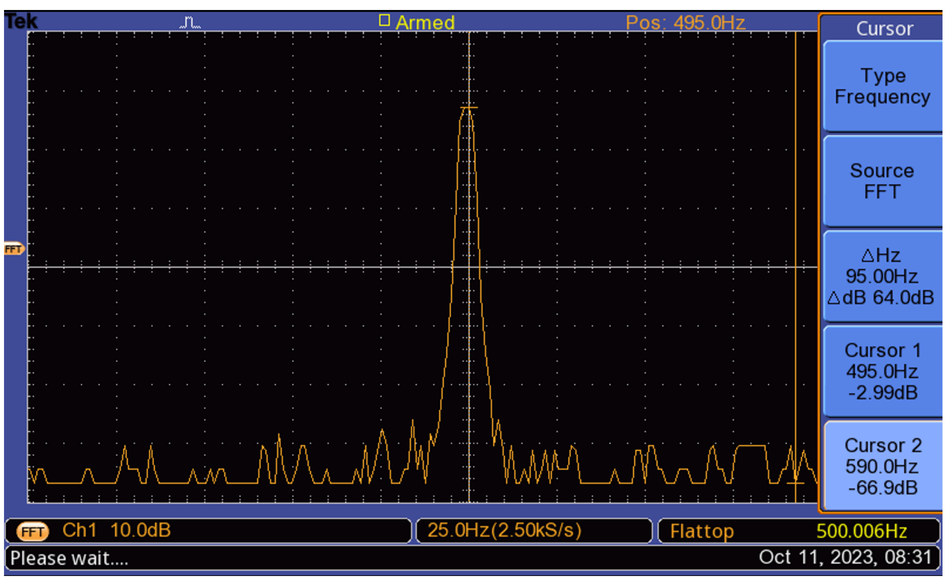


Hard copy of time domain signal

The peak-peak voltage is ~2V and the frequency is ~500Hz

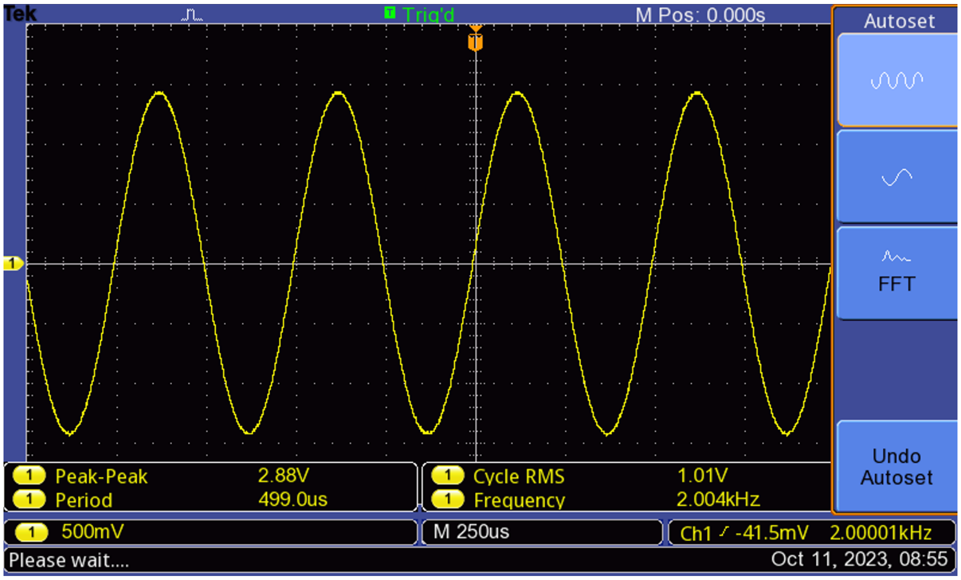


Complete FFT spectrum

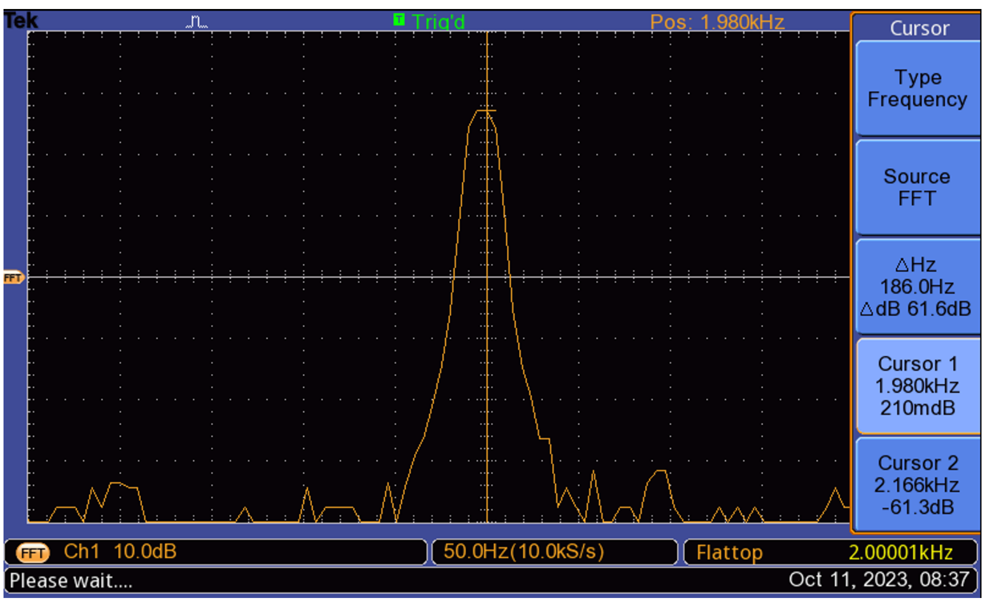


Zoomed-in FFT spectrum with cursor

As evident in the figure above, the peak is at ~500Hz, which is expected, however the Flattop and Cursor values are not the same because the scroll wheel has discrete ‘clicks’ and we could not get it to go to the exact top of the peak.



2kHz frequency sinusoidal signal with 0dB spectrum peak in time domain



2kHz frequency sinusoidal signal with 0dB spectrum peak in frequency domain

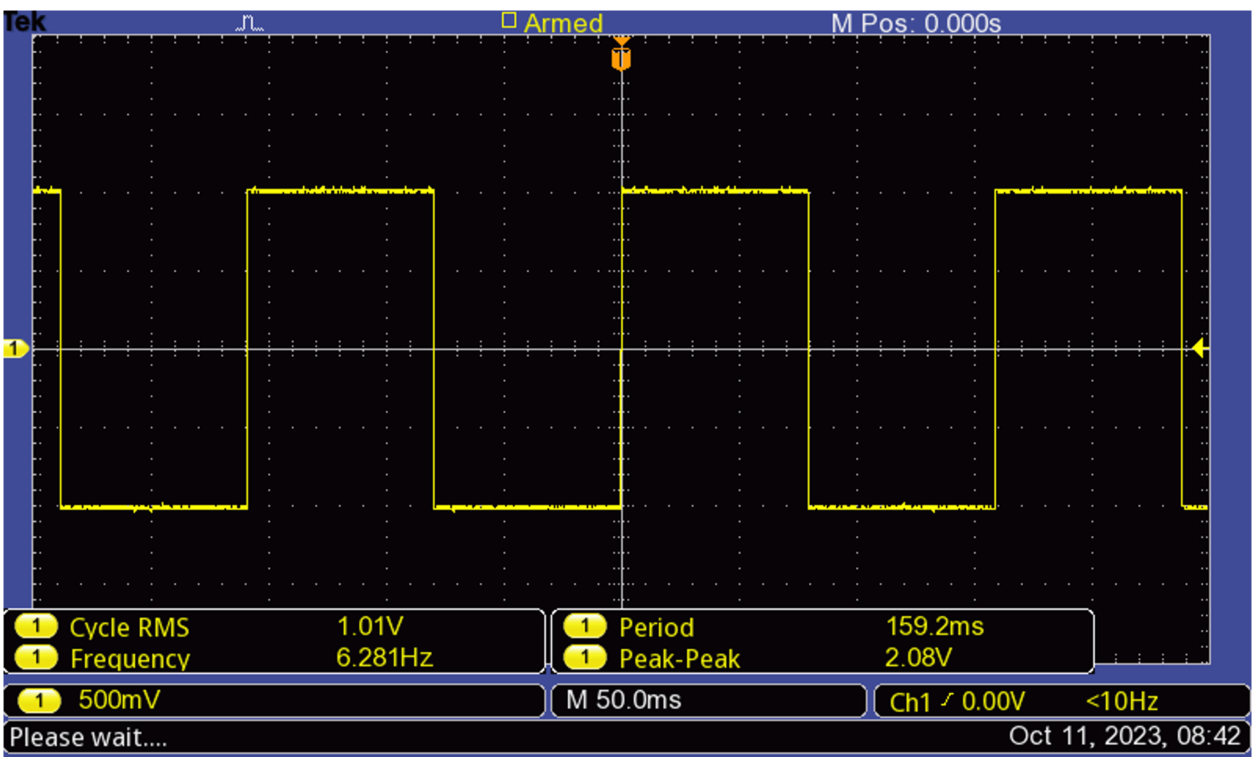
As can be seen in the cursor, the peak is at 210mdB which is approximately 0. The amplitude is 2.88V

Part 2 - FFT of square wave

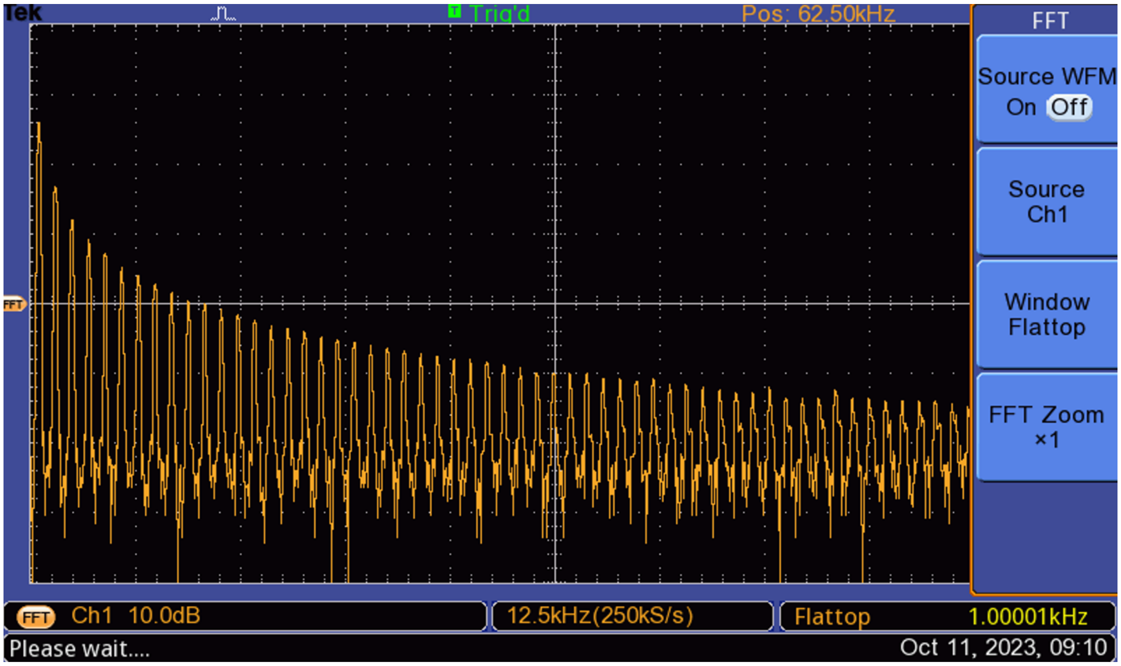
Experimental Setup and Procedure

1. Use the function generator to generate a square wave having 1 ms period, 2 Vpp amplitude, and no offset. Check the properties with the measure function. Take a hard copy.
2. Obtain the FFT spectrum. Instead of using the time base (sec/div) control to accurately measure the frequency components, use the FFT zoom control that provides a zoom factor up to 10 and use the cursors to determine the amplitudes and the frequency of the fundamental and the first four harmonics. Take hard copies of the FFT signal.
3. Obtain the FFT spectrum for 20 % duty cycles. Determine the amplitudes of the fundamental frequency and the first four harmonics. Take hardcopies of the signal in time and frequency domains.

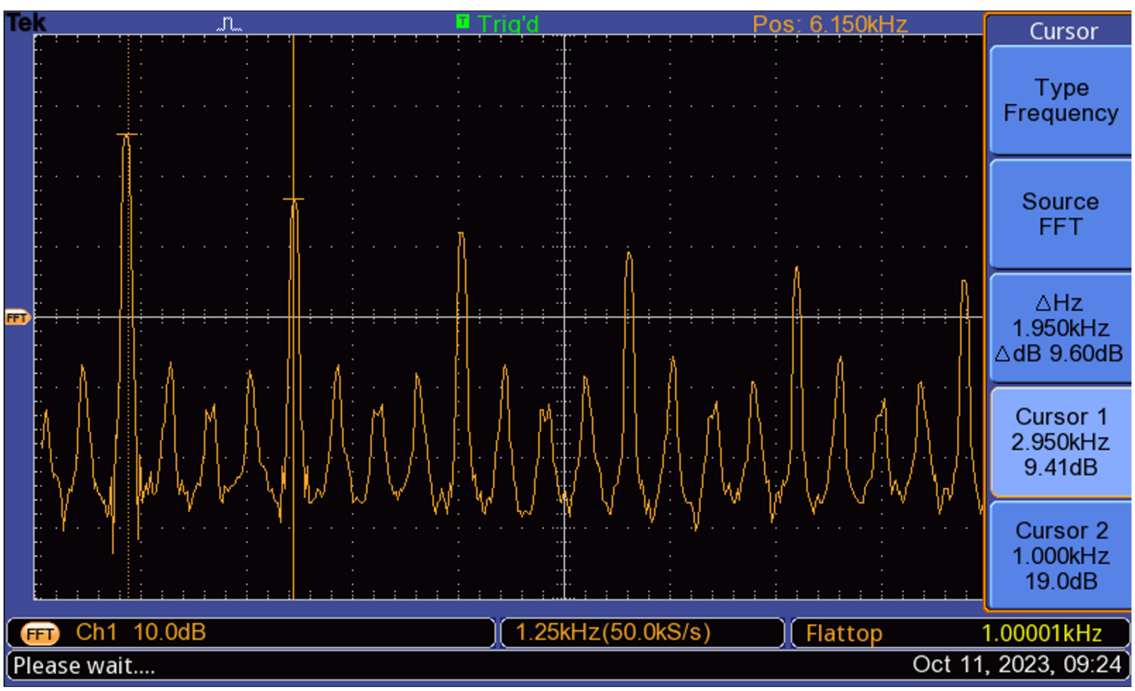
Results



$2V_{pp}$ rectangular wave



Full FFT spectrum of Square wave

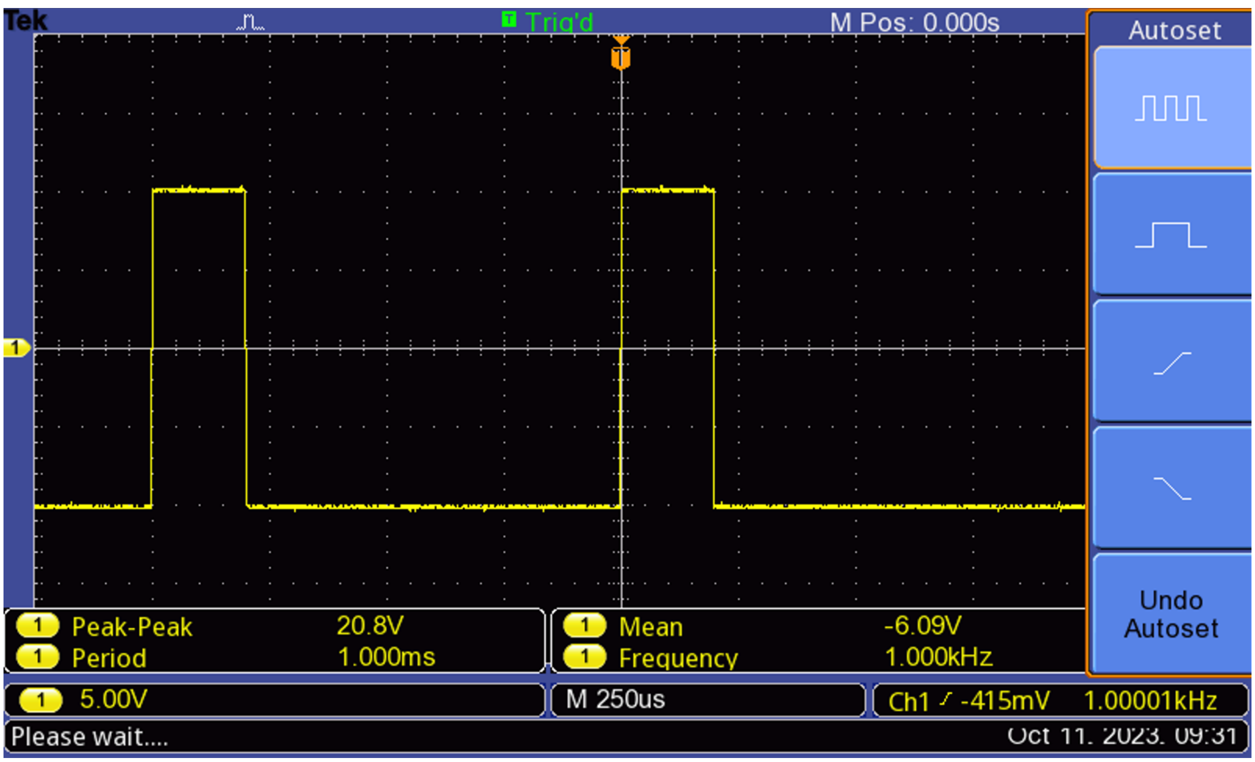


Spectrum with FFT zoom factor of 10 with cursors on the fundamental and first harmonics

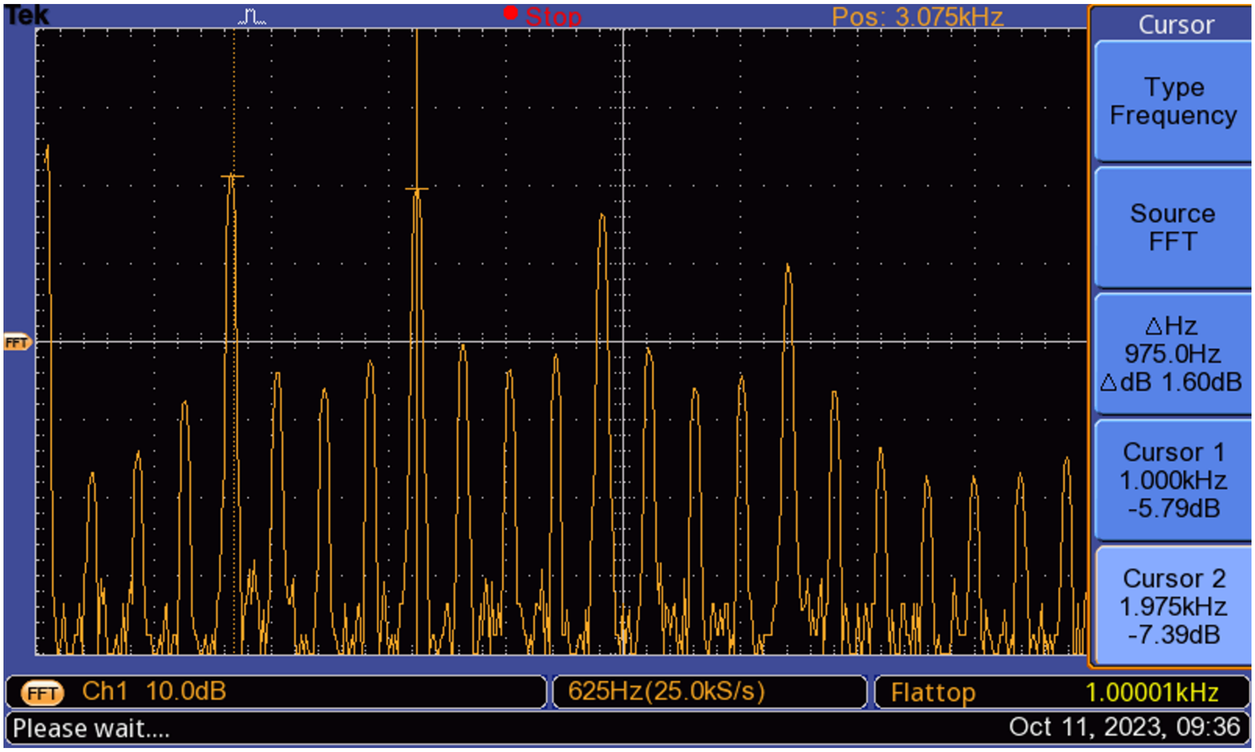
The following results were measured using the cursor on the oscilloscope:

Harmonic	Frequency (Hz)	Amplitude (dB)
Fundamental	1000	19.000
First	2950	9.410
Second	4950	5.010
Third	6850	2.210
Fourth	8900	-0.189

Then we obtained the spectrum for 20% duty cycles:



Square wave (20% duty cycles)



Spectrum of Square wave (20% duty cycles)

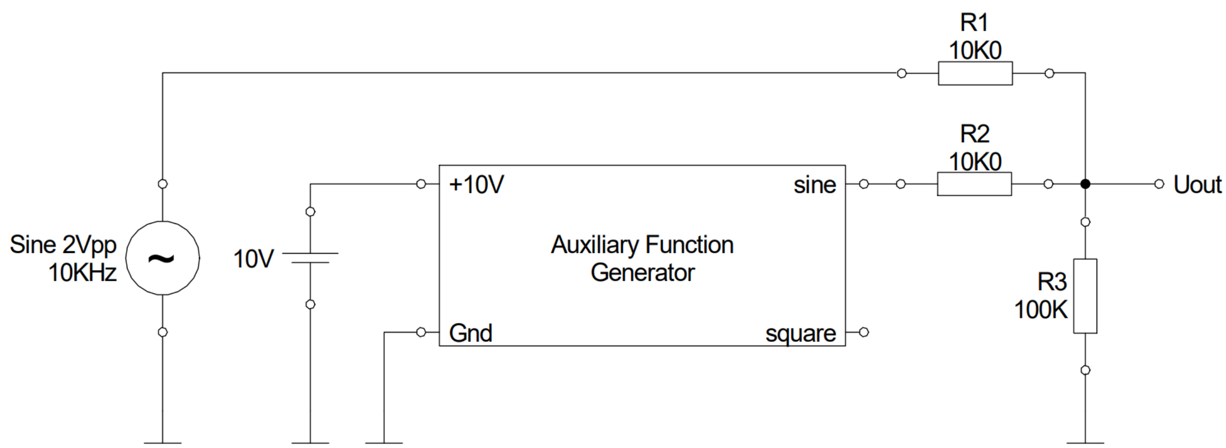
The following results were measured using the cursor on the oscilloscope:

Harmonic	Frequency (Hz)	Amplitude (dB)
Fundamental	1000	-5.8
First	1975	-7.4
Second	2975	-10.5
Third	3975	-17.3
Fourth	4950	-44.1

Part 3 - FFT of a multiple-tone sinusoidal wave

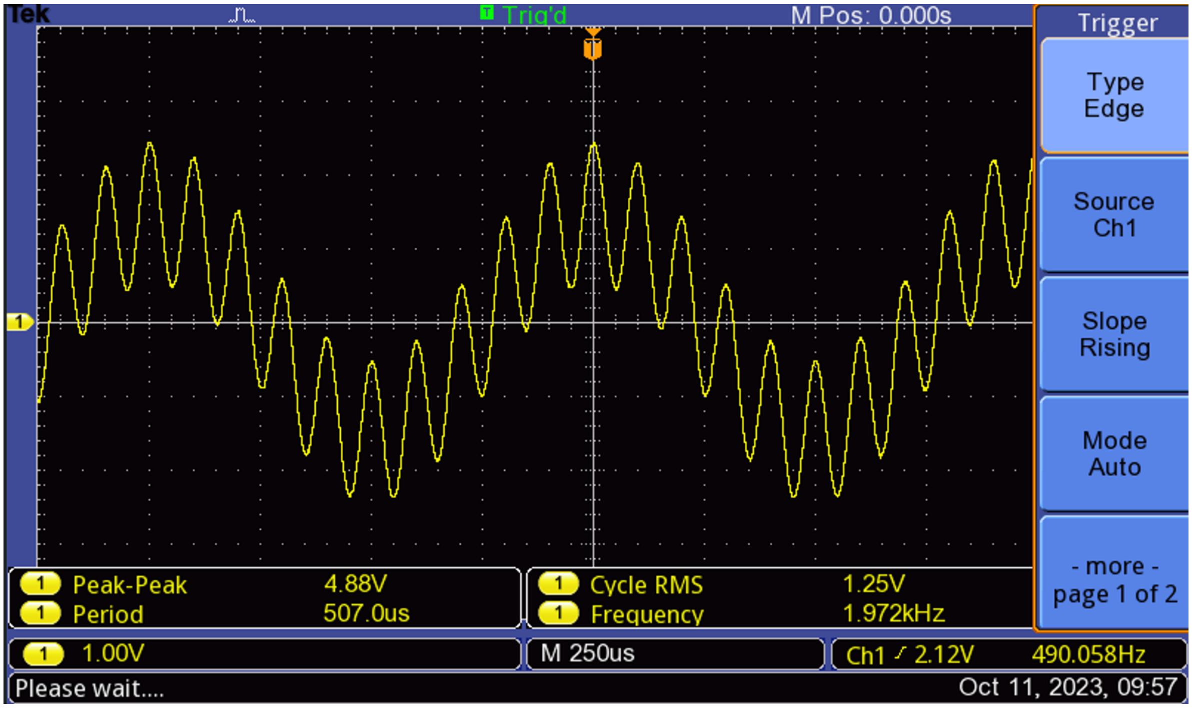
Experimental Setup and Procedure

1. Combine the signal from the sine output of the auxiliary signal generator and a 2 Vpp, 10 kHz sinusoidal wave from the Agilent signal generator. Use the following circuit.

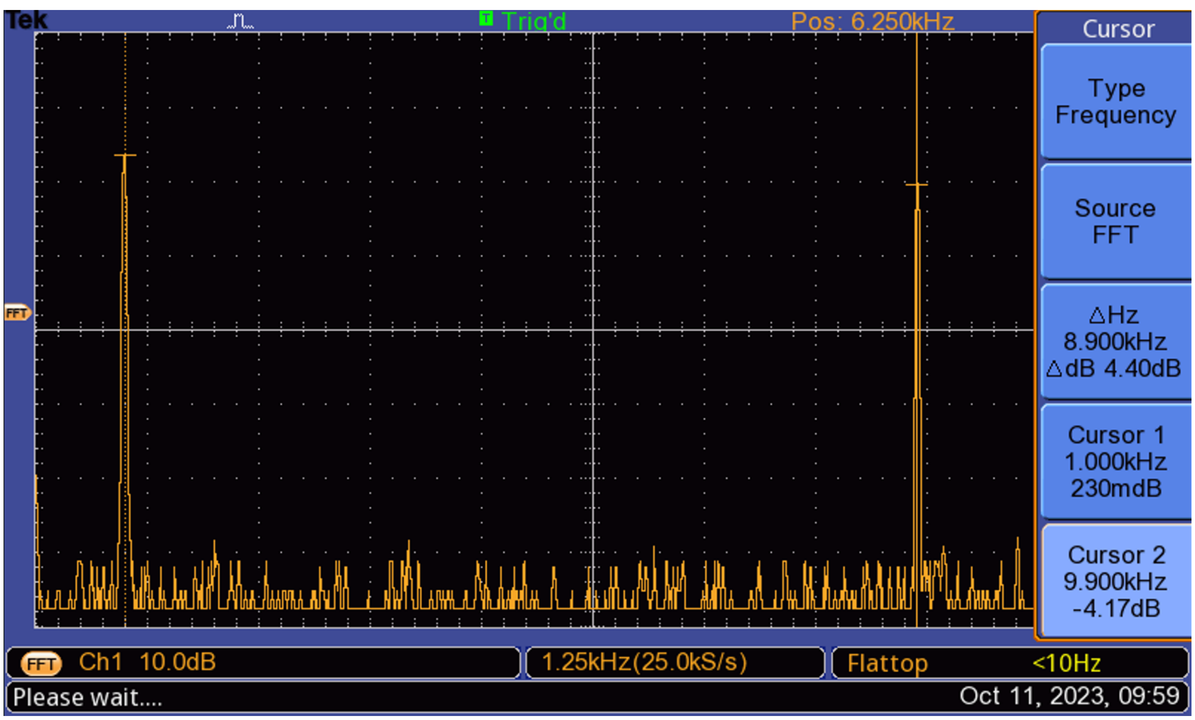


2. Take a hard copy of the signal in time domain.
3. Take a hard copy of the FFT spectrum of the signal.

Results



Multi-tone sine wave signal in time domain



Multi-tone sine wave signal in frequency domain

Evaluation

Problem 1 - FFT of Single Tone sinusoidal wave

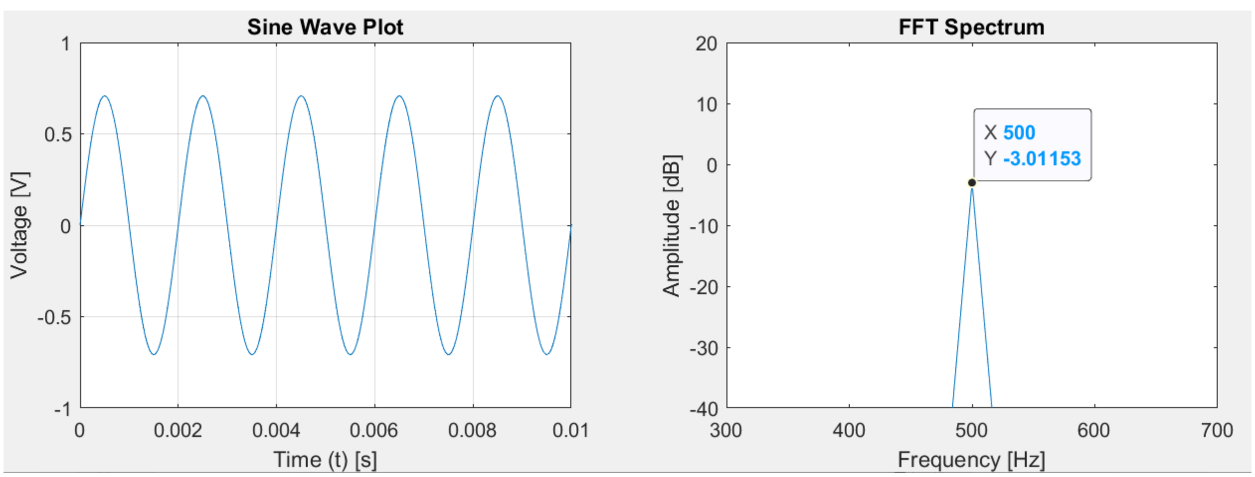
What is the reference value of the oscilloscope for 0dB?

The reference value of the oscilloscope at 0dB is 1 V_{RMS}

Use Matlab to calculate the expected FFT spectra for the parameters given in part 1.1. Is the calculated spectra consistent with the measured spectra?

```
%Generating and plotting Sine wave
t = 0:0.00001:0.05;
f = 500;
w0 = 2*pi*f;
A = sqrt(2)/2;
s = A*sin(w0*t);
subplot(1,2,1);
plot(t, s);
xlabel('Time (t) [s]');
ylabel('Voltage [V]');
title('Sine Wave Plot');
xlim([0,0.01]);
ylim([-1, 1]);
grid on;
```

```
%Implementing and plotting FFT
fs=100000;
f_nyq=fs/2;
L=length(s);
s_fft=fft(s);
s_fft=2*abs(s_fft/L);
db_s_fft=db(s_fft);
db_s_fft=db_s_fft(1:L/2+1);
f_domain=linspace(0,f_nyq,L/2+1);
subplot(1,2,2);
plot(f_domain,db_s_fft);
xlabel("Frequency [Hz]");
ylabel("Amplitude [dB]");
title("FFT Spectrum");
xlim([300,700]);
ylim([-40,20]);
```



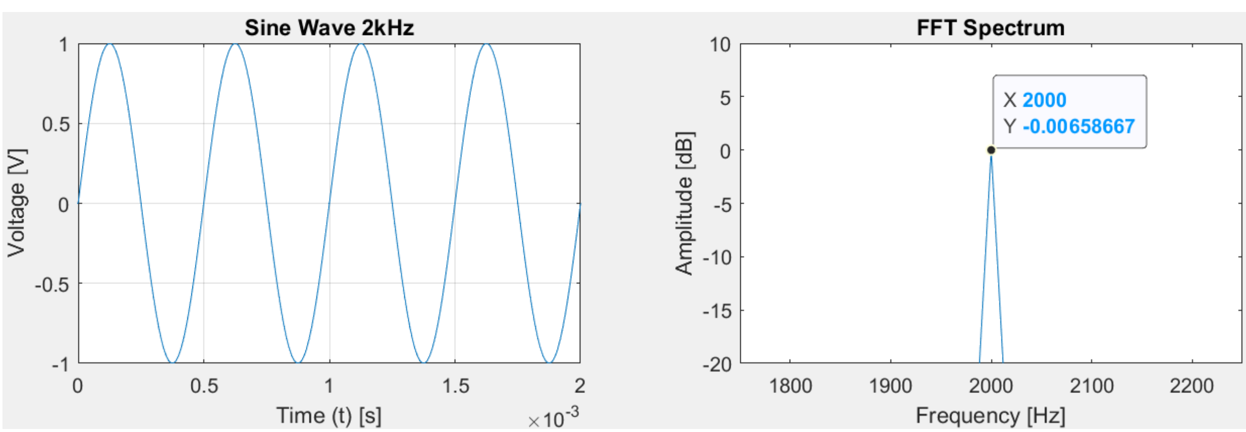
	Frequency (Hz)	Amplitude (dB)
Measured	495	-2.99
Calculated	500	-3.01

The MATLAB spectrum reveals a peak at 500 Hz with an amplitude of -3.01 dB, closely mirroring the calculated values, where the measured amplitude registers at -2.99 dB at 495 Hz. The minimal error (less than 1%) suggests consistency between the calculated and measured spectra. However, aligning the cursor precisely with the peak is challenging, and the slight inaccuracy may stem from potential oscilloscope measurement errors. Moreover, there is a possibility of a deviation in the signal generator's operational accuracy, contributing to the observed discrepancies. The overall close agreement between the calculated and measured spectra indicates a reliable representation, despite the small margin of error.

Use Matlab to calculate the expected FFT spectra for the parameters given in part 1.3. Is the calculated spectra consistent with the measured spectra?

```
%Generating and plotting Sine wave
fs=100000;
t = 0:1/fs:0.05;
f = 2000;
w0 = 2*pi*f;
A = 1;
s = A*sin(w0*t);
subplot(1,2,1);
plot(t, s);
xlabel('Time (t) [s]');
ylabel('Voltage [V]');
title('Sine Wave 2kHz');
xlim([0,0.002]);
ylim([-1, 1]);
grid on;
```

```
%Implementing and plotting FFT
f_nyq=fs/2;
L=length(s);
s_fft=fft(s);
s_fft=2*abs(s_fft/L);
db_s_fft=db(s_fft);
db_s_fft=db_s_fft(1:L/2+1);
f_domain=linspace(0,f_nyq,L/2+1);
subplot(1,2,2);
plot(f_domain,db_s_fft);
xlabel("Frequency [Hz]");
ylabel("Amplitude [dB]");
title("FFT Spectrum");
xlim([1750,2250]);
ylim([-20,10]);
```



	Frequency (Hz)	Amplitude (dB)
Measured	1980	0.21
Calculated	2000	-0.01

In the spectrum from MATLAB the peak is at 2000 Hz, where it has an amplitude of ~(-0.01dB). However, the measured amplitude at 1980 Hz is 210mdB. Even though both values are considered sufficiently close to the intended 0dB peak, the difference in dB is very large, which means the calculated spectrum is not consistent with the measured spectrum in dB. The large error may be due to the limitations of the equipment. Some faults are, again, the error in measurement (precision) of the oscilloscope or the small deviation in the precision of the cursor as it's very difficult to align the cursor with the peak. There might also be some fault in the signal generator's accuracy.

Compare the results from Matlab with the measured values. Discuss the differences.

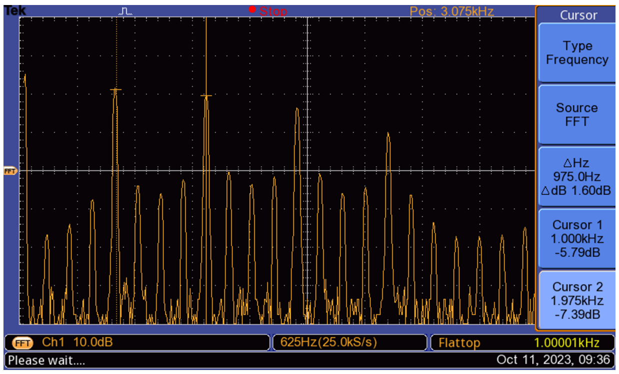
In general, the spectra derived from calculations exhibit more distinct and pointed peaks compared to those obtained through the oscilloscope, where the peaks display a slight curvature. Despite some differences between the calculated and measured data, these variations are minimal and fall within an acceptable range, affirming a reasonable level of agreement between the two sets of values. The discrepancies in readings can be attributed to limitations in equipment precision. Measurement errors by the oscilloscope and slight deviations in the accuracy of the signal generated by the signal generator are potential factors contributing to these differences. Additionally, challenges in achieving precise cursor alignment on the oscilloscope's peak further underscore the limitations in equipment precision.

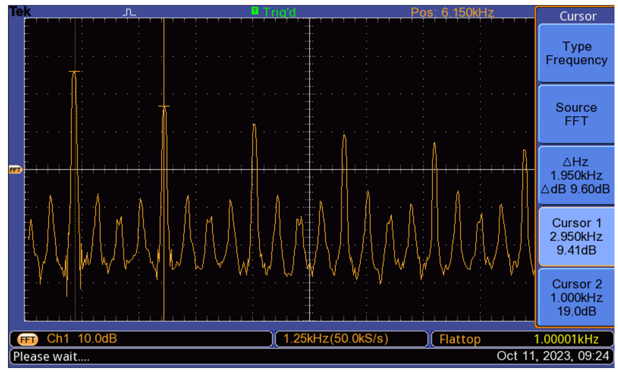
Problem 2 - FFT of Square Wave

For frequency domain measurements, the frequency scale needs to be expanded in order to accurately measure the frequency components. This could be done with the time base (sec/div) control. What is the effect of doing this on the measured bandwidth?

Improving the precision of frequency component measurement involves adjusting the time base control (measured in seconds per division), facilitating an expansion of the frequency scale. The configuration of the Fast Fourier Transform (FFT) resolution is also achievable through manipulation of the time base control. The alteration of the time base results in a corresponding modification of the sampling rate, influencing the observation of the signal. Extending the time base to amplify the signal leads to a decrease in the measured bandwidth, while a reduction in the time base widens the bandwidth. This highlights the reciprocal relationship between the time base and bandwidth.

Use the hardcopies taken to discuss the effect of changing the duty cycle on the FFT results.

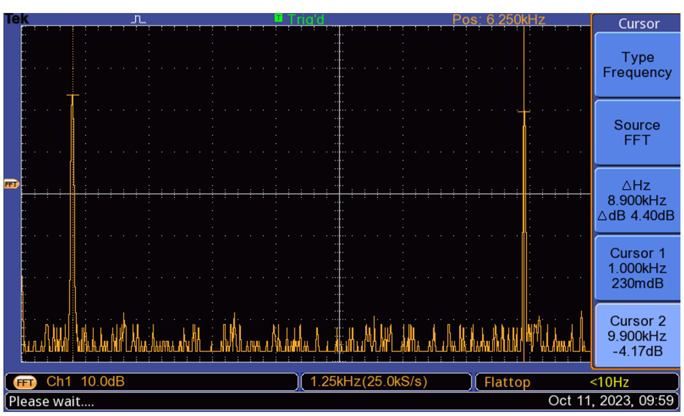




The duty cycle represents the ratio of pulse width to the period. As evident, an increase in the duty cycle results in an increase in the DC offset value, and conversely, a decrease in the duty cycle leads to a reduction in the DC offset. In the spectrum, the square wave generated with a 20% duty cycle (on the left) exhibits a narrower width compared to the one without a duty cycle (on the right). Consequently, the inclusion of a duty cycle in the FFT spectrum results in a narrowing of the width.

Problem 3 - FFT of multiple-tone sinusoidal wave

Use the hardcopy of the spectrum and discuss the linearity of the FFT.



The spectrum of a sinusoidal wave in FFT appears as an impulse. When a second sinusoidal wave from the auxiliary signal generator is introduced and their FFTs are combined, two impulses are observed. This demonstrates that the addition of functions in the time domain corresponds to their addition in the Fourier domain, thereby affirming the linearity property of the Fourier Transform. This means FFT maintains the linearity property. As we are adding the two sinusoids, we can see two peaks at ~1KHz and ~10KHz.

Conclusion

The experiments undertaken were centered on delving into signal analysis in the Frequency or Fourier Domain, specifically focusing on sinusoidal signals and square waves. The application of Fast Fourier Transform (FFT) to these signals was explored, and their characteristics were examined and documented. Using an oscilloscope and a signal generator, FFT results were generated for sinusoidal and square wave signals at various frequencies. The FFT spectra were analyzed under different peak-to-peak voltage settings for sinusoidal signals and various duty cycle values for square waves. Combining signals from the main and auxiliary signal generators, we investigated the linearity in Fourier transform, validating our measurements against MATLAB-generated plots.

Throughout the experiment, practical challenges were encountered, such as the unattainability of a 0 dB spectrum peak due to inherent systematic errors in the instruments. Notably, sinusoidal signals in the time domain corresponded to impulses in the Fourier domain, while square waves exhibited harmonics in the frequency domain with diminishing amplitudes, known as the sinc() function. Alterations in the duty cycle were found to impact the direct current (DC) offset value in the FFT spectrum. The experiment also included a test of the linearity property of the Fourier Transform, with results validated.

In this lab, a comprehensive understanding of the Fourier Transform was achieved, encompassing conceptual calculations and practical applications using MATLAB. Sinusoids and square waves were constructed in MATLAB and analyzed through the fft() function. The generation of sinusoids and square waves using the signal generator, along with the study of their spectra using the oscilloscope, provided valuable insights. The FFT feature, measure function, and cursors facilitated precise measurements for experimentation. Combining two sinusoids using the auxiliary signal generator showcased the linearity property of the Fourier Transform.

Comparisons between prelab information and experimental results aided in understanding the theoretical underpinnings of the Fourier transform. The limitations of equipment in different situations were grasped, along with strategies for optimizing equipment use. In the evaluation, theoretical evidence was reconstructed and comparisons were made, identifying subtle nuances impacting experimental outcomes and understanding conditions for validating experimental data using theoretical means.

References

Pagel, Uwe. *CO-520-B Signals and Systems Lab Manual*. 2023.

Appendix

Prelab - Sampling

Problem 1 - The Sampling Theorem

Analog signals are usually passed through a low-pass filter prior to sampling. Why is this necessary?

Aliasing, caused by high-frequency elements in an analog signal, can render distinct signals indistinguishable from one another. To address this problem, the removal of high-frequency components is achieved by passing the signals through a low-pass filter.

What is the minimum sampling frequency for a pure sine wave input at 3KHz? Assume that the signal can be completely reconstructed.

To achieve thorough and precise sampling and reconstruction of the signal, the sampling frequency needs to be at least twice the input frequency. For a pure sine wave, the minimum required sampling frequency is calculated as 2*3 kHz, resulting in 6 kHz.

What is the Nyquist frequency?

Nyquist frequency is the minimum frequency above which a signal must be sampled to fully recover the original input signal.

What are the resulting frequencies for the following input sinusoids 500Hz, 2.5KHz, 5KHz and 5.5KHz if the signals are sampled by a sampling frequency of 5KHz?

The resulting frequency after sampling a function is given by:

$$f_{result} = \left| f_0 - f_s \cdot nint\left(\frac{f_0}{f_s}\right) \right|$$

Where f_0 is the input frequency, f_s is the sampling frequency, and nint gives the nearest integer. The resulting frequencies can be calculated using the formula.

Input Frequency (f_0) (Hz)	Resulting frequency (f_{result}) (Hz)
500	500
2500	2500
5000	0
5500	500

Mention three frequencies of signal that alias to a 7Hz signal. The signal is sampled by a constant 30 Hz sampling frequency.

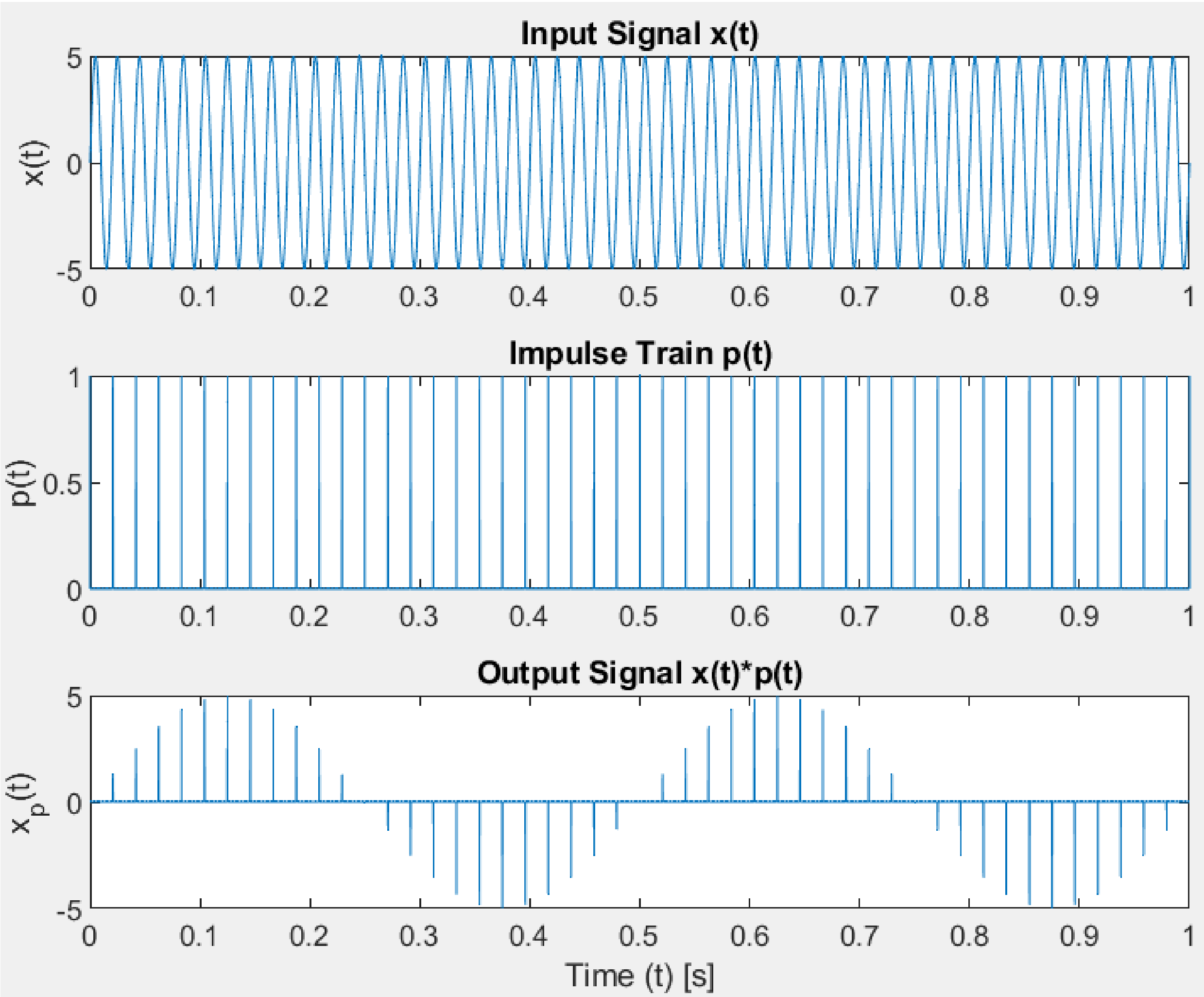
The frequencies should be 7Hz more than a multiple of the sampling frequency, hence the three frequencies can be 37Hz, 67Hz, and 97Hz.

Problem 2 - Impulse Train Sampling and Real Sampling

The input signal x(t) is given by a sine function, with an amplitude of 5 V peak and a frequency of 50 Hz. The sampling signal p(t) is represented by a unity impulse train. Use an overall sampling rate of 100 k samples/s) for the whole problem.

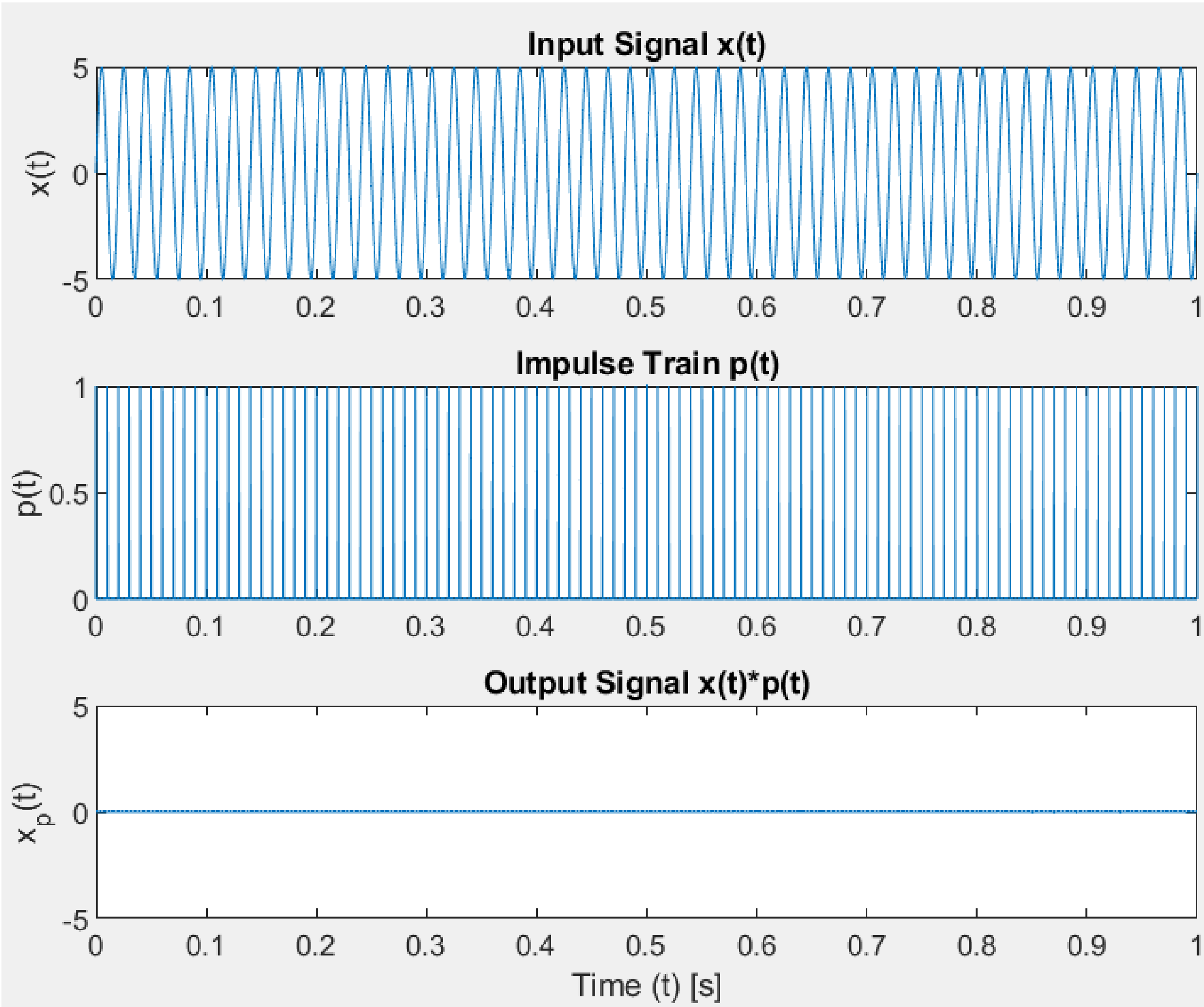
Carry out simulations for the following cases, and use the command subplot to visualize the continuous signal x(t), the sampling signal p(t), and the result for each of these cases:

a. **Under Sampling (use 48 Hz)**


Input Signal x(t)
x(t)
5
0
-5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Impulse Train p(t)
p(t)
1
0.5
0
Output Signal x(t)*p(t)
$x_p(t)$
Time (t) [s]

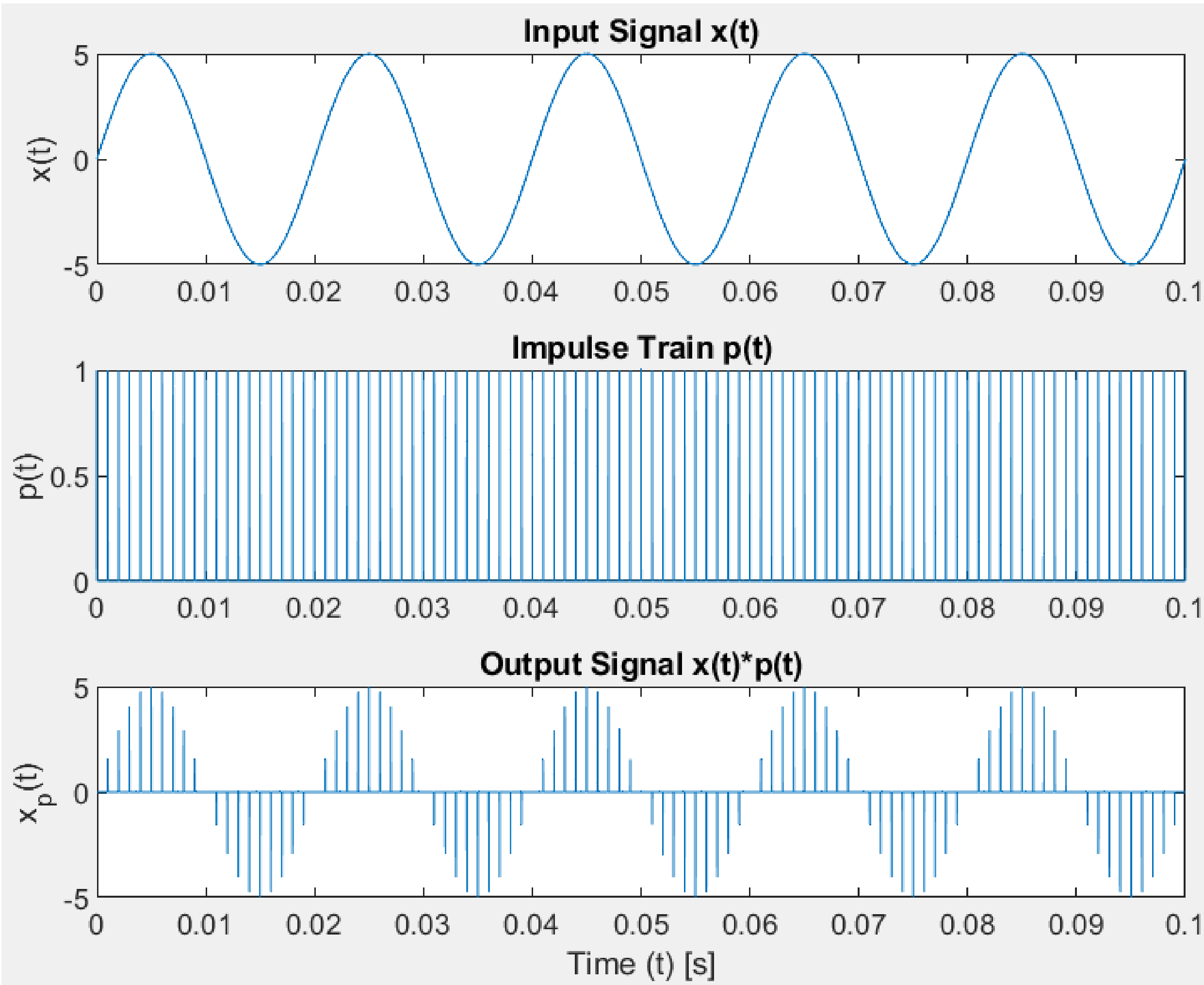
```
%Under Sampling 48Hz
Fs=100000;
fs=48;
t=0:1/Fs:1;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
impt=(1+square(2*pi*fs*t,0.1))/2;
subplot(3,1,2);
plot(t,impt);
title("Impulse Train p(t)");
ylabel("p(t)");
subplot(3,1,3);
xp=x.*impt;
plot(t,xp);
title("Output Signal x(t)*p(t)");
xlabel("Time (t) [s]");
ylabel("x_{p}(t)");
ylim([-5,5]);
```

b. Nyquist Sampling



```
%Nyquist Sampling 2*50=100Hz
Fs=100000;
fs=100;
t=0:1/Fs:1;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
impt=(1+square(2*pi*fs*t,0.1))/2;
subplot(3,1,2);
plot(t,impt);
title("Impulse Train p(t)");
ylabel("p(t)");
subplot(3,1,3);
xp=x.*impt;
plot(t,xp);
title("Output Signal x(t)*p(t)");
xlabel("Time (t) [s]");
ylabel("x_{p}(t)");
ylim([-5,5]);
```

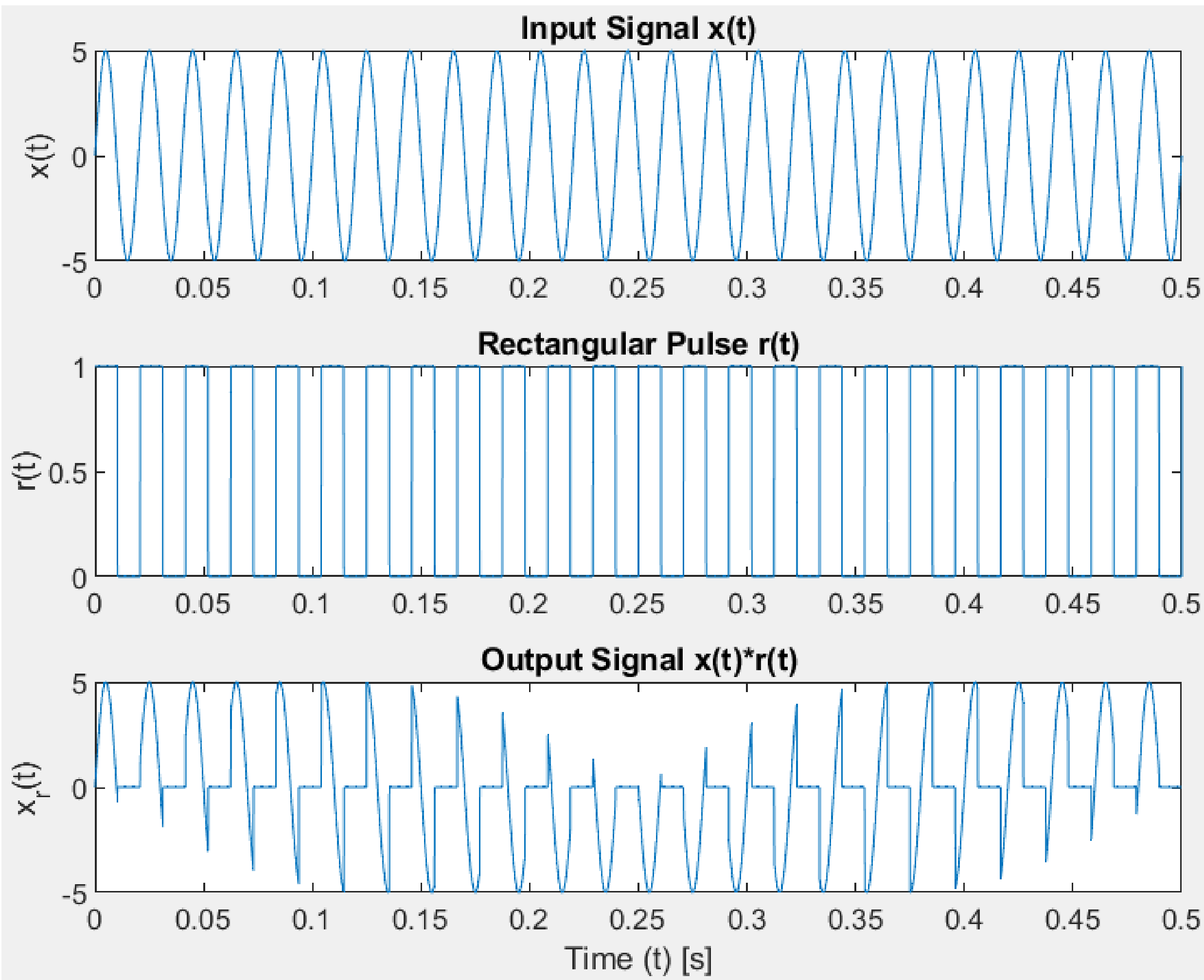
c. Over Sampling (use 1000Hz)



```
%Over Sampling 1000Hz
Fs=100000;
fs=1000;
t=0:1/Fs:0.1;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
impt=(1+square(2*pi*fs*t,0.1))/2;
subplot(3,1,2);
plot(t,impt);
title("Impulse Train p(t)");
ylabel("p(t)");
subplot(3,1,3);
xp=x.*impt;
plot(t,xp);
title("Output Signal x(t)*p(t)");
xlabel("Time (t) [s]");
ylabel("x_{p}(t)");
ylim([-5,5]);
```

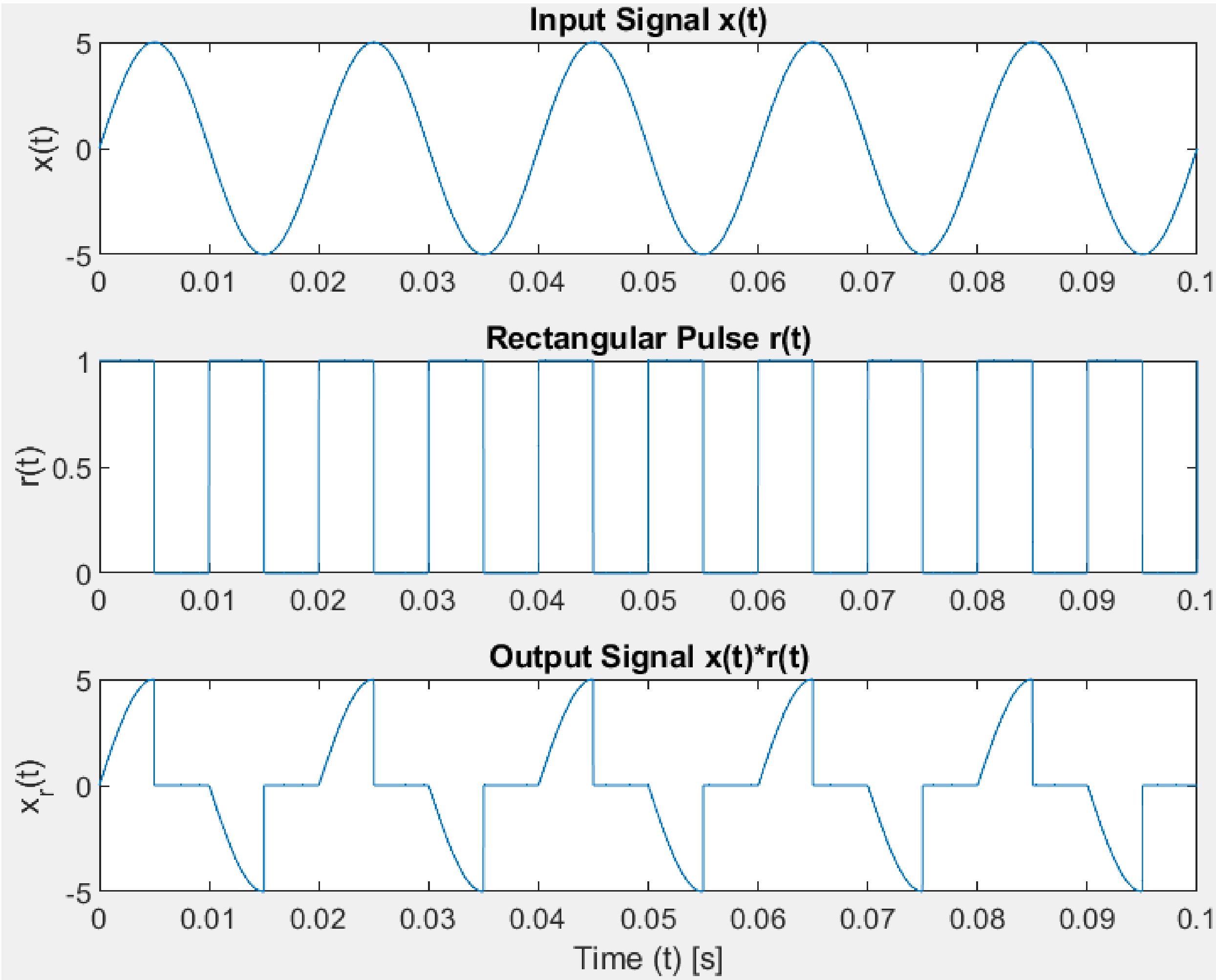
The signal x(t) should be sampled by a rectangular pulse train. Modify the sampling function p(t), so that the width of the sampling pulse is 50% of the sampling period. Carry out simulations for the following cases: (a) Under Sampling (b) Nyquist Sampling (c) Over Sampling. Use the same sampling rates and the same plot setup as before.

Under Sampling:



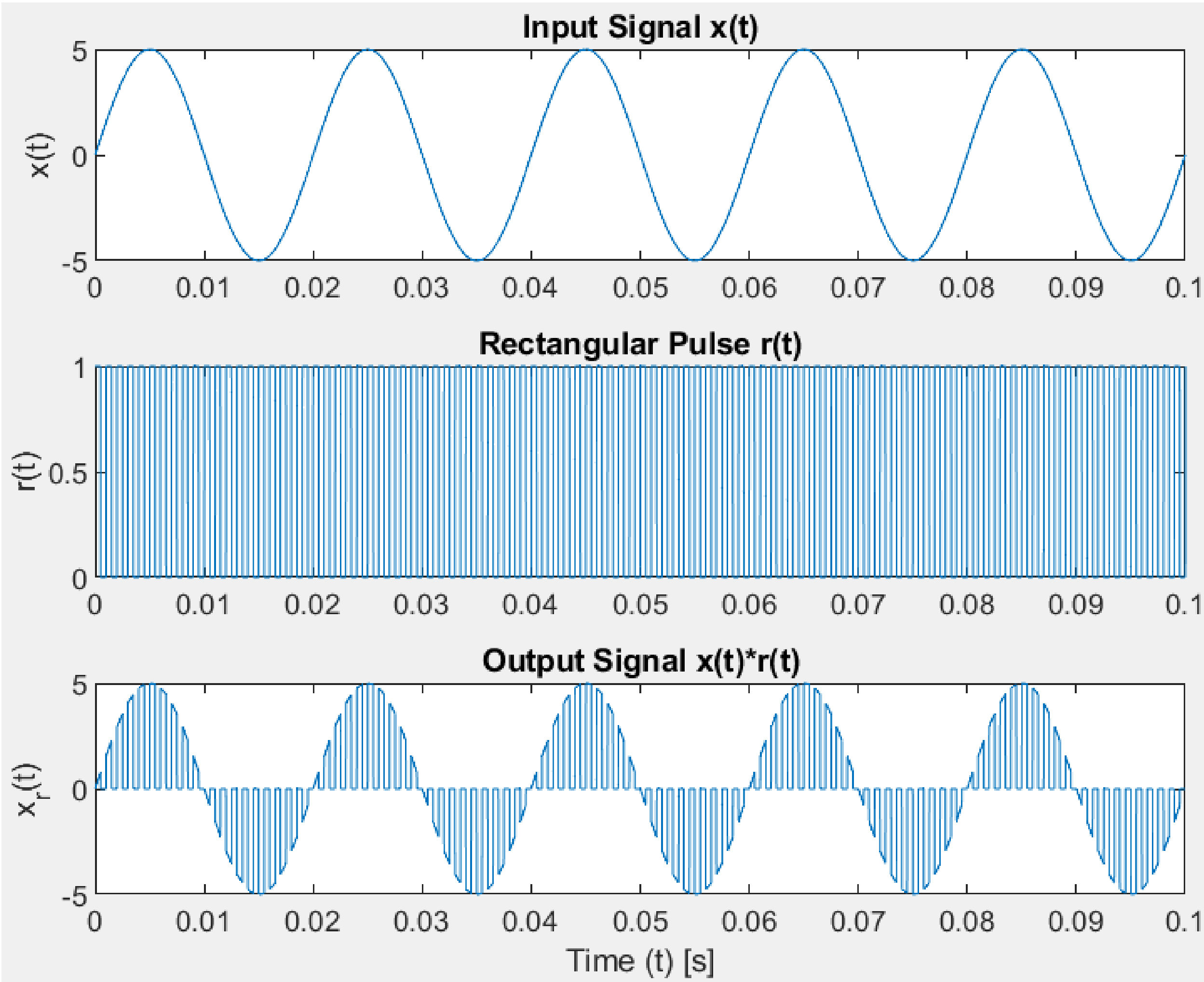
```
%Under Sampling 48Hz
Fs=100000;
fs=48;
t=0:1/Fs:0.5;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
rec=[0.1,50];
for train = 1: length(rec)
    for rate = 1: length(fs)
        rect=max(square(2*pi*fs(rate)*t,rec(train)),0);
        subplot(3,1,2);
        plot(t,rect);
        title("Rectangular Pulse r(t)");
        ylabel("r(t)");
        subplot(3,1,3);
        xp=x.*rect;
        plot(t,xp);
        title("Output Signal x(t)*r(t)");
        xlabel("Time (t) [s]");
        ylabel("x_{r}(t)");
        ylim([-5,5]);
    end
end
```

Nyquist Sampling:



```
%Nyquist Sampling 2*50=100Hz
Fs=100000;
fs=100;
t=0:1/Fs:0.1;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
rec=[0.1,50];
for train = 1: length(rec)
    for rate = 1: length(fs)
        rect=max(square(2*pi*fs(rate)*t,rec(train)),0);
        subplot(3,1,2);
        plot(t,rect);
        title("Rectangular Pulse r(t)");
        ylabel("r(t)");
        subplot(3,1,3);
        xp=x.*rect;
        plot(t,xp);
        title("Output Signal x(t)*r(t)");
        xlabel("Time (t) [s]");
        ylabel("x_{r}(t)");
        ylim([-5,5]);
    end
end
```

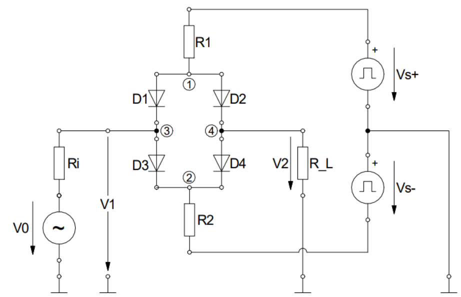
Over Sampling:



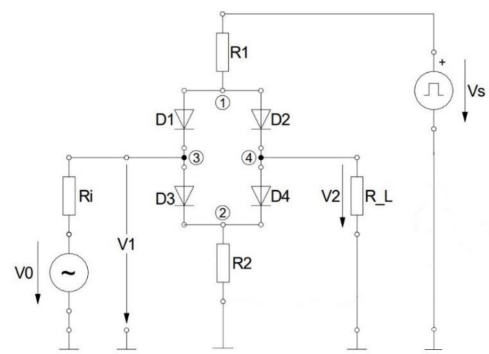
```
%Over Sampling 1000Hz
Fs=100000;
fs=1000;
t=0:1/Fs:0.1;
w0=2*pi*50;
x=5*sin(w0*t);
subplot(3,1,1);
plot(t,x);
title("Input Signal x(t)");
ylabel("x(t)");
rec=[0.1,50];
for train = 1: length(rec)
    for rate = 1: length(fs)
        rect=max(square(2*pi*fs(rate)*t,rec(train)),0);
        subplot(3,1,2);
        plot(t,rect);
        title("Rectangular Pulse r(t)");
        ylabel("r(t)");
        subplot(3,1,3);
        xp=x.*rect;
        plot(t,xp);
        title("Output Signal x(t)*r(t)");
        xlabel("Time  (t)  [s]");
        ylabel("x_{r}(t)");
        ylim([-5,5]);
    end
end
```

Problem 3: Sampling using a Sampling bridge

Modify the circuit in the figure below in such a way that a single sampling source can be used to sample the input signal.



Sketch the modified circuit.



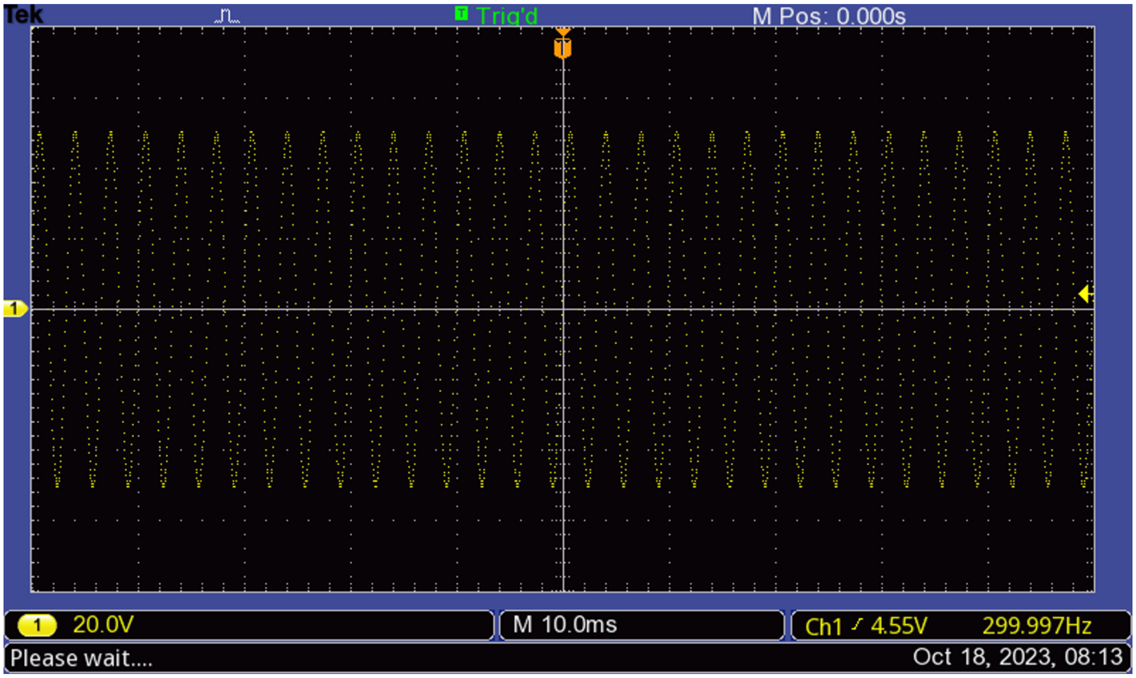
Explain the operation of the modified circuit.

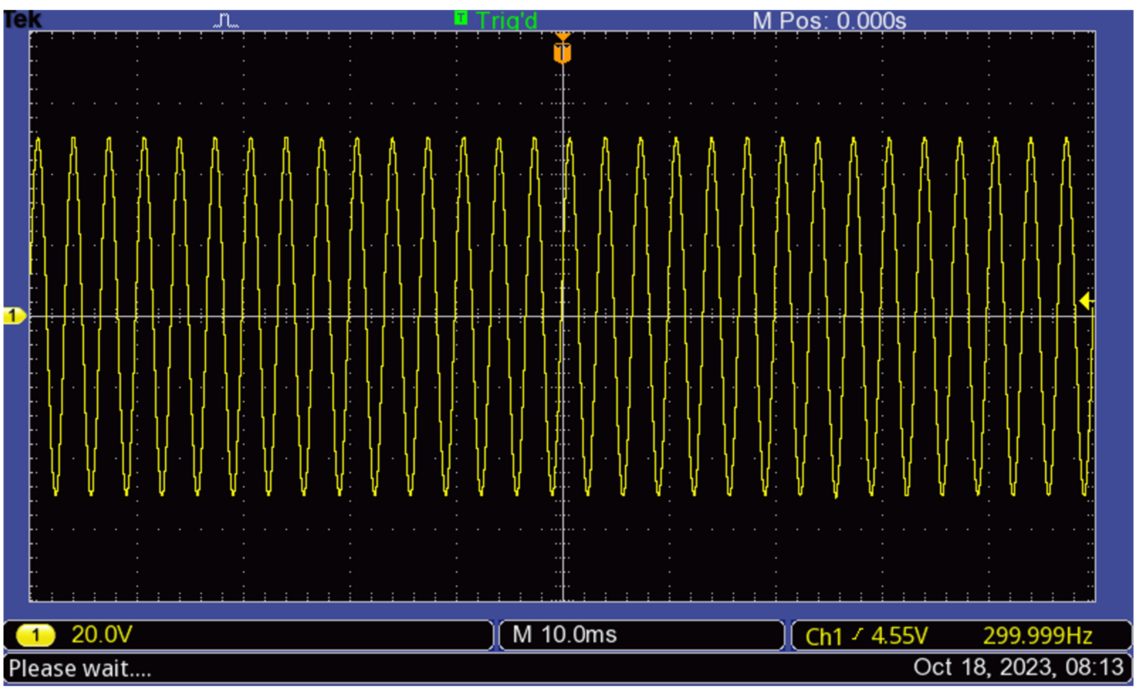
One distinction between the initial circuit and the depicted circuit above lies in the ground configuration. In the original circuit, signal variations occur around 0V. In contrast, the altered circuit incorporates an offset, causing a mathematical shift in the resulting signal by the offset value. This shift is a consequence of eliminating the Vs-. The removal of the negative sampling source implies that only one sampling source remains to sample the signal.

Results - Sampling

Part 1 - Digital Sampling Oscilloscope

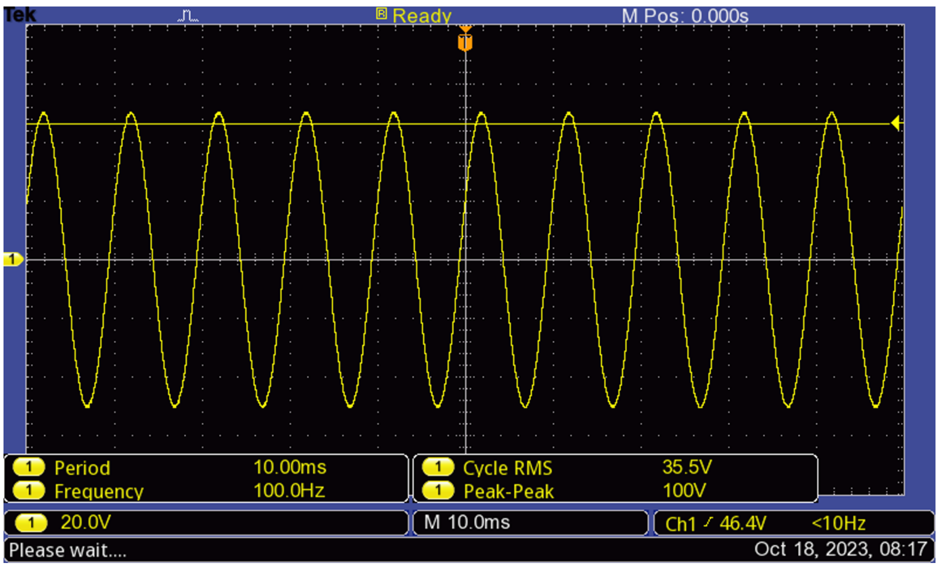
1. **Demonstrate that the graph on the oscilloscope screen consists of single points**

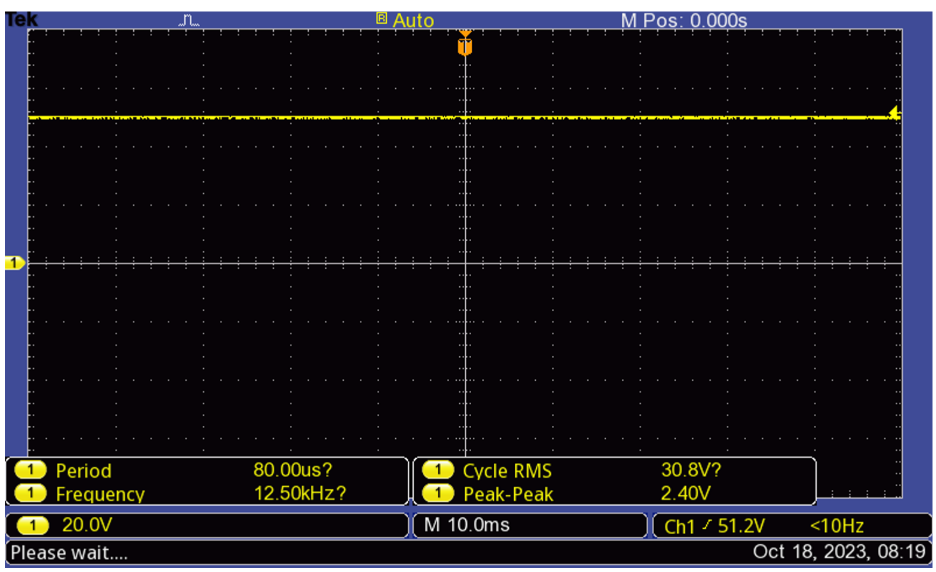


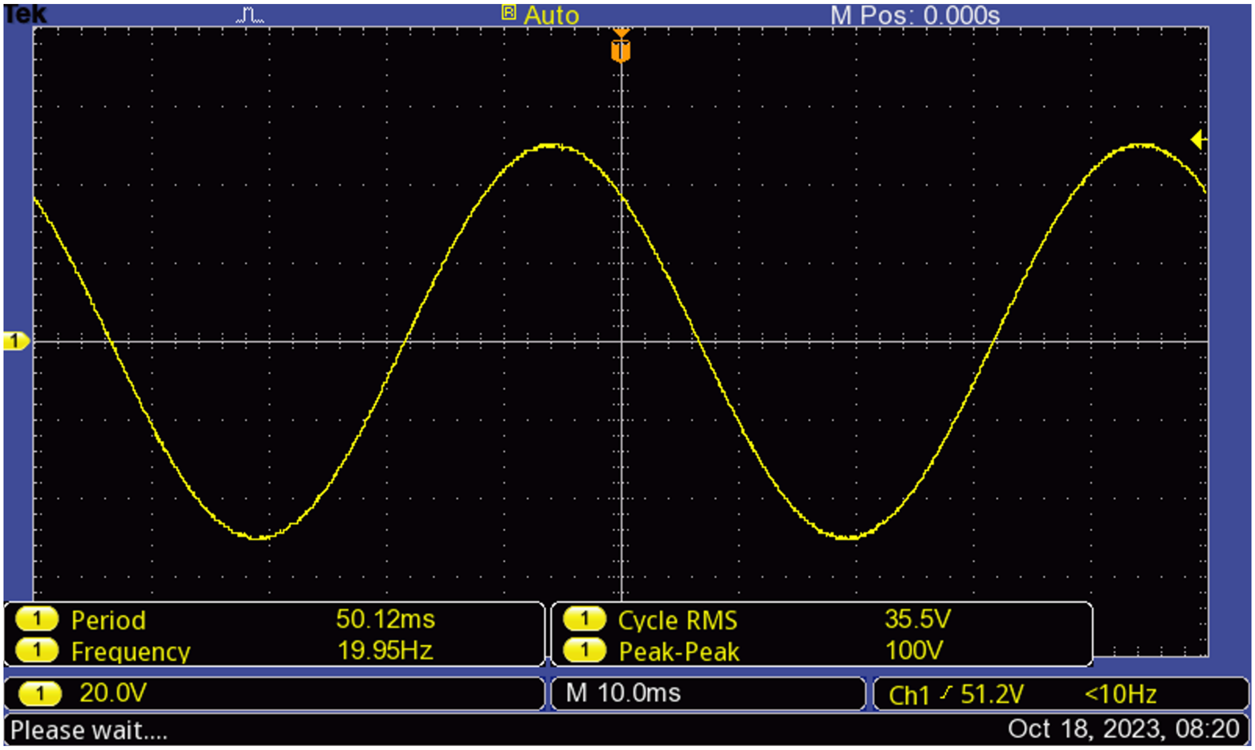


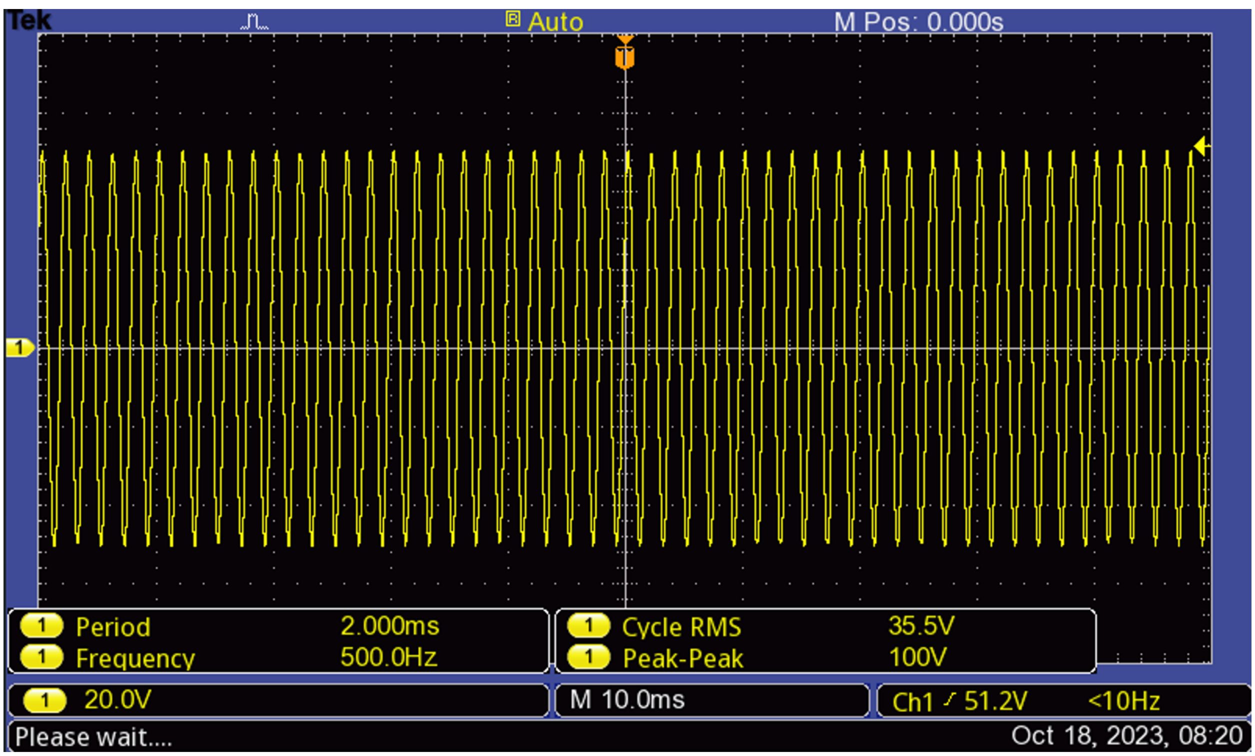
2. What happens when the input signal exceeds the Nyquist frequency? Keep the sampling rate at the oscilloscope at 25 kS/s. Set the frequency at the generator to 24900 Hz, 25000 Hz, 25020 Hz, and 25500 Hz. Use the 'MEASURE' function to measure the alias frequency! Take a hardcopy for every step.

Hard copies taken for 24.9kHz, 25kHz, 25.02kHz, and 25.5kHz respectively with alias frequency

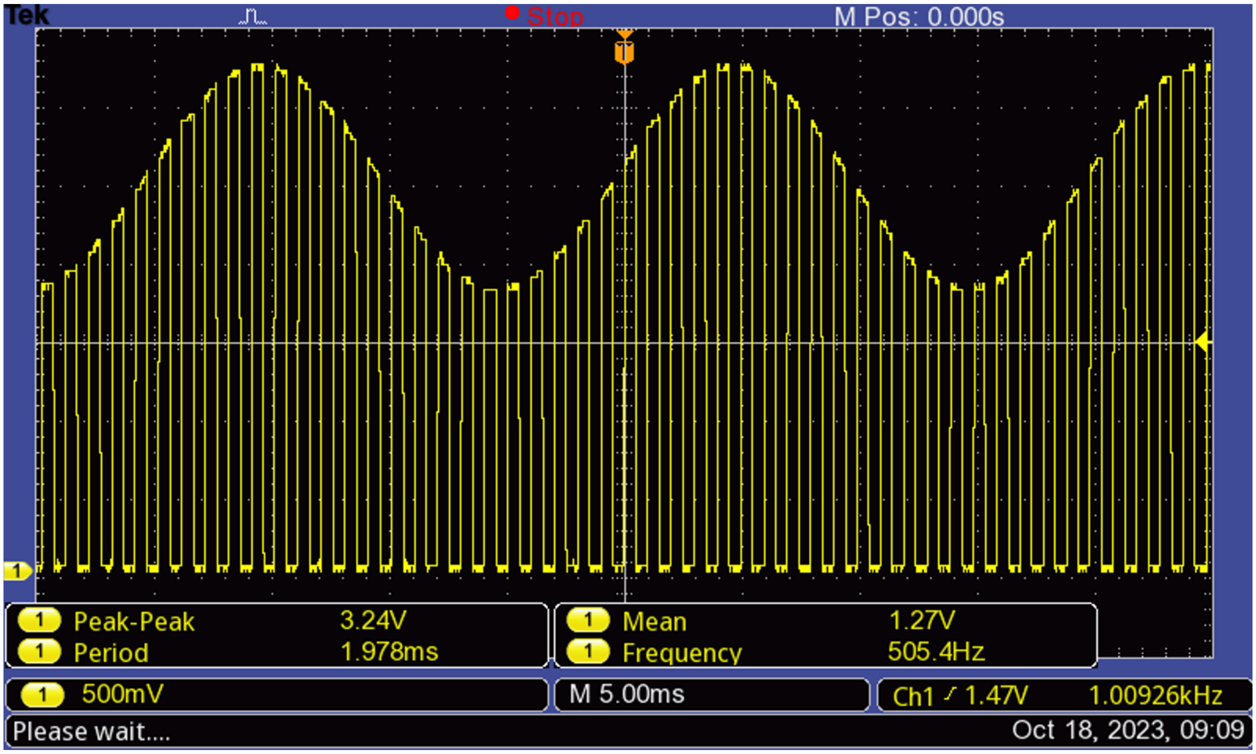


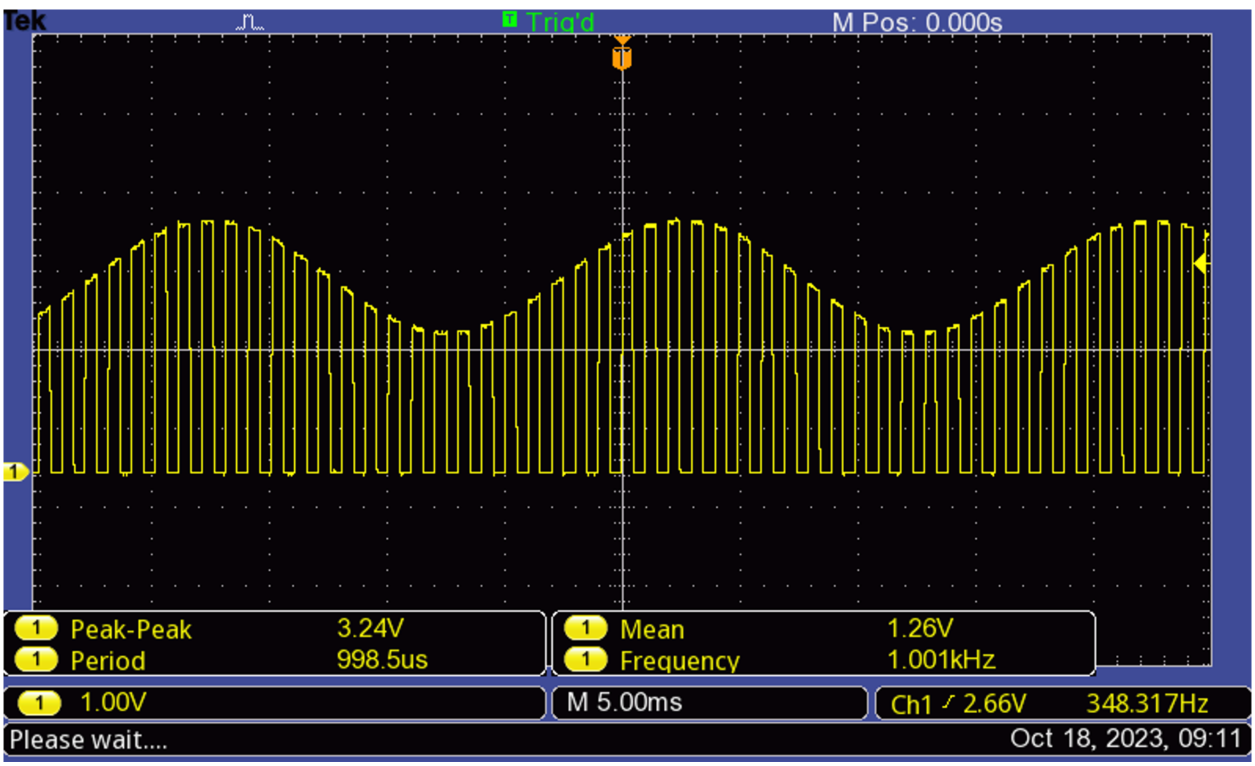


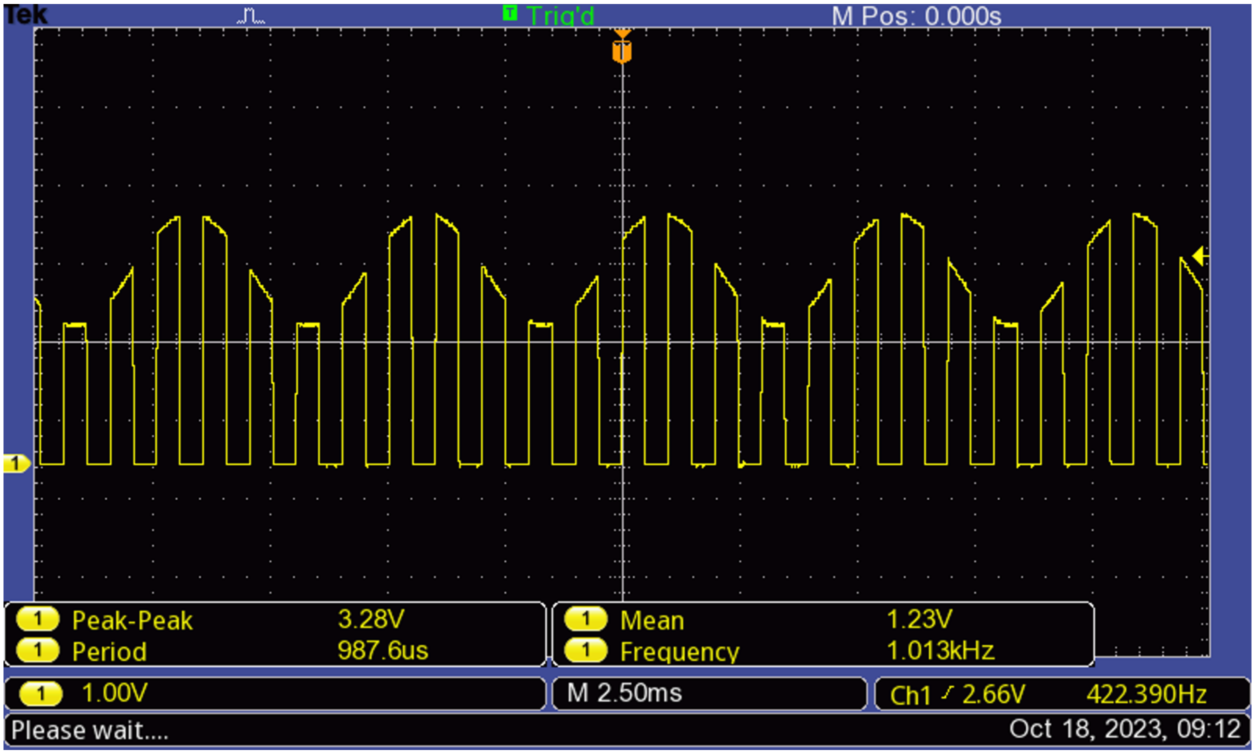

Tek
Auto
M Pos: 0.000s
Period 50.12ms
Frequency 19.95Hz
Cycle RMS 35.5V
Peak-Peak 100V
20.0V
M 10.0ms
Ch1 51.2V <10Hz
Please wait....
Oct 18, 2023, 08:20

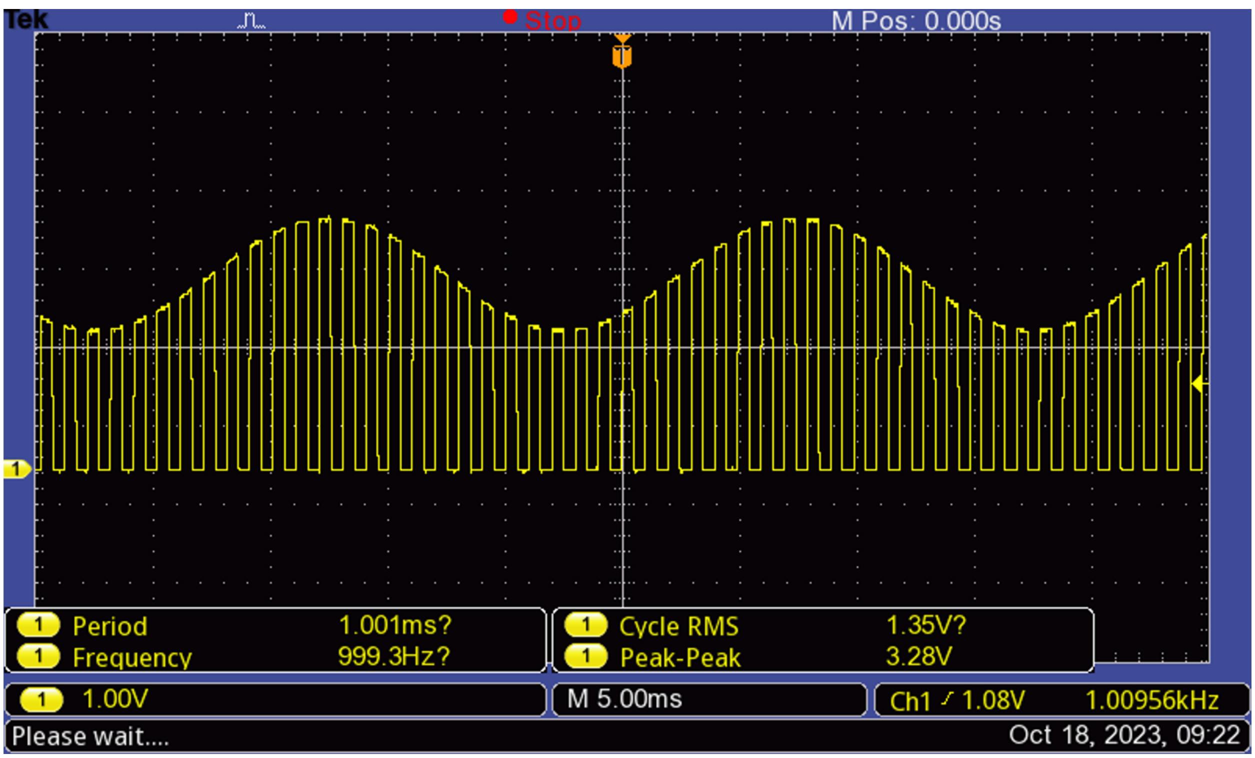

Tek
Auto
M Pos: 0.000s
Period 2.000ms
Frequency 500.0Hz
Cycle RMS 35.5V
Peak-Peak 100V
20.0V
M 10.0ms
Ch1 51.2V <10Hz
Please wait....
Oct 18, 2023, 08:20

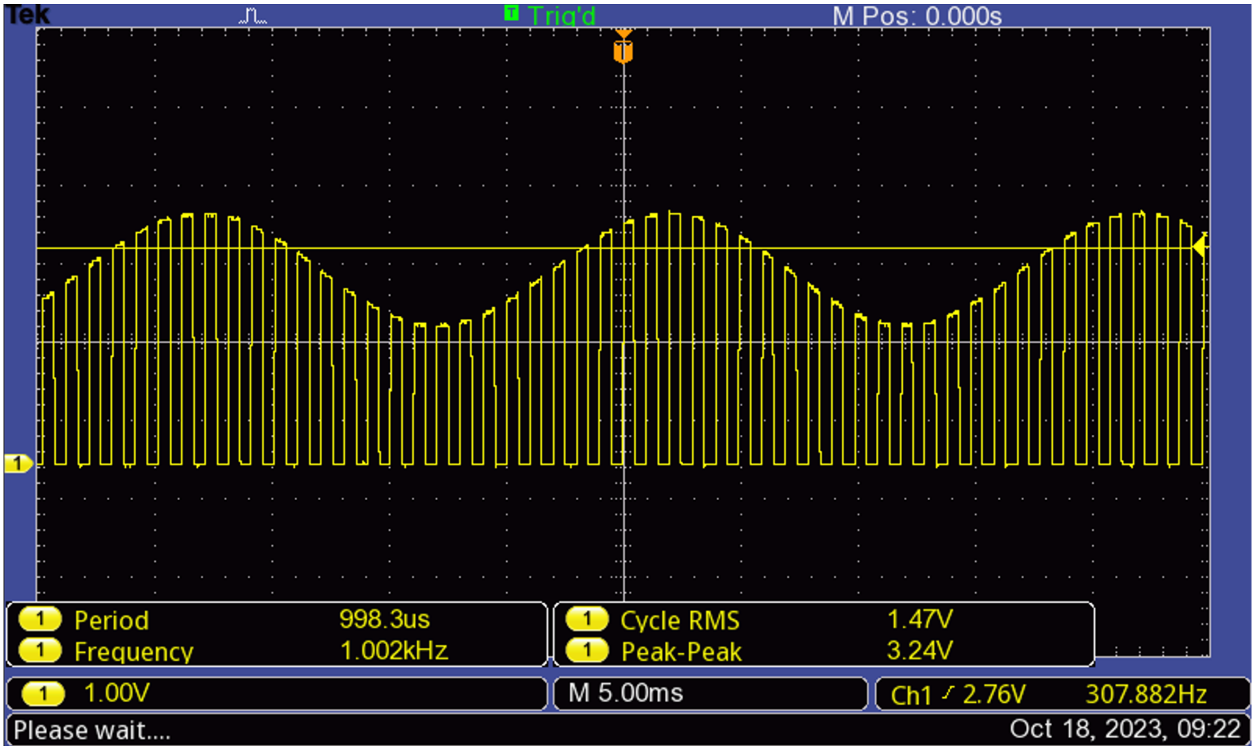
Part 2 - Sampling using a sampling bridge

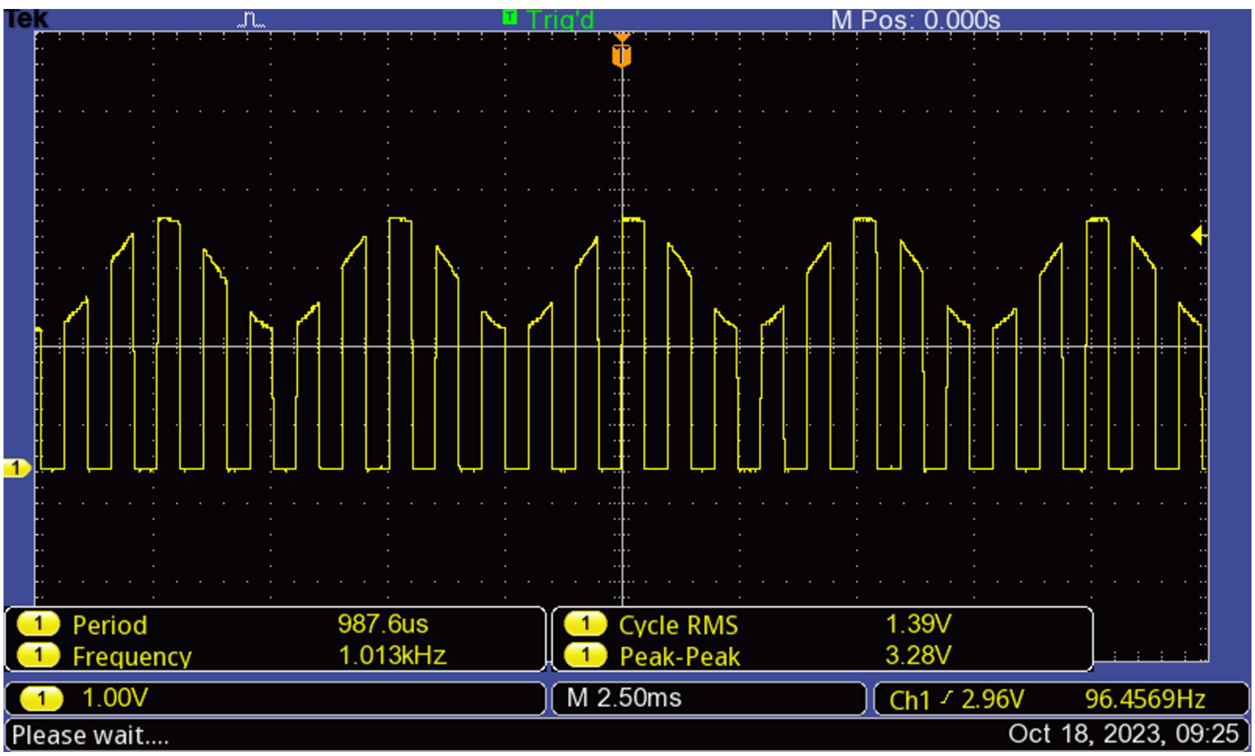





Tek
Trig'd
M Pos: 0.000s
Peak-Peak 3.28V
Period 987.6us
Mean 1.23V
Frequency 1.013kHz
1.00V
M 2.50ms
Ch1 ⁄ 2.66V 422.390Hz
Please wait....
Oct 18, 2023, 09:12


Tek
Stop
M Pos: 0.000s
Period 1.001ms?
Frequency 999.3Hz?
Cycle RMS 1.35V?
Peak-Peak 3.28V
1.00V
M 5.00ms
Ch1 ⁄ 1.08V 1.00956kHz
Please wait....
Oct 18, 2023, 09:22


Tek
Trig'd
M Pos: 0.000s
Period 998.3us
Frequency 1.002kHz
Cycle RMS 1.47V
Peak-Peak 3.24V
1.00V
M 5.00ms
Ch1 ∫ 2.76V 307.882Hz
Please wait....
Oct 18, 2023, 09:22


Tek
Trig'd
M Pos: 0.000s
Period 987.6us
Frequency 1.013kHz
Cycle RMS 1.39V
Peak-Peak 3.28V
1.00V
M 2.50ms
Ch1 ∫ 2.96V 96.4569Hz
Please wait....
Oct 18, 2023, 09:25