Constructor University Bremen

Lab Report 2: The Fourier Series and The Fourier Transform

Fall Semester 2024

Author: **Idriz Pelaj**

Experiment Conducted By:
Mr. Idriz Pelaj, Mr. Getuar Rexhepi

Conducted on: **October 11th, 2023**

1. Prelab

1.1 Problem 1: Decibels

1. Given $x(t) = 5\cos(2\pi 1000t)$

   - What is the amplitude and the V_{pp} of the signal?
     The amplitude of the signal is $5V$ and the V_{pp} is $2 \cdot 5V = 10V$.
   - What is the V_{rms} of the signal?
     The V_{rms} of a sinusoidal signal is given by

$$V_{rms} = \frac{V_A}{2\sqrt{2}} = \frac{5V}{\sqrt{2}} = 3.535V \tag{1.1}$$

   - What is the amplitude of the spectral peak in dBVrms? The amplitude of the spectral peak in dBVrms is given by the following relation:

$$A_{dBV_{rms}} = 20\log_{10}(V_{rms}) = 20\log_{10}(3.535V) = 10.9dBV_{rms} \tag{1.2}$$

2. Given a square wave of $1V_{pp}$ the voltage changes between -0.5V to 0.5V

   - What is the signal amplitude in V_{rms}? To find the RMS value of a square wave, the following relation is used:

$$V_{rms} = \sqrt{\frac{1}{T}\left[\int_0^T f(t)^2 dt\right]} \tag{1.3}$$

     The square wave can be specified as a piecewise function:

$$f(t) = \begin{cases} -0.5 & \text{if } 0 \leq t \leq \frac{T}{2} \\ 0.5 & \text{if } \frac{T}{2} \leq t \leq T \end{cases} \tag{1.4}$$

Performing the integral over the function,

$$\begin{aligned}
V_{rms} &= \sqrt{\frac{1}{T}\left[\int_0^{T/2}(-0.5)^2 dt + \int_{T/2}^{T}(0.5)^2 dt\right]} \\
&= \sqrt{0.25 \cdot \frac{1}{T}\left(\frac{T}{2} + T - \frac{T}{2}\right)} \\
&= \sqrt{0.25}V \\
&= 0.5V
\end{aligned} \tag{1.5}$$

- What is the amplitude in dBVrms?

$$V_{dB_{rms}} = 20\log_{10}(V_{rms}) = 20\log_{10}(0.5V) = -6.02dBV_{rms} \tag{1.6}$$

1.2 Problem 2: Determination of Fourier Series Coefficients

1. Determine the Fourier series coefficients up to the 5th harmonic of the function $f(t) = 4t^2$

   To determine the fourier series coefficients, first recognize that the function is an even function by the very fact that it is a polynomial of an even degree. Knowing this, the relations used to compute the fourier series coefficients are:

$$\begin{aligned}
a_n &= \frac{4}{T}\int_{-T}^{T} t^2 \cos(n\omega_0 t)dt \\
a_0 &= \frac{1}{T}\int_0^T t^2 dt \\
b_n &= 0
\end{aligned} \tag{1.7}$$

   Where it is known that $T = 1$ due to the interval being given as $[-0.5, 0.5]$. Furthermore, $\omega_0 = \frac{2\pi}{T} = 2\pi$. By using the help of an integration table, the integral for a_n is

$$a_n = 4 \cdot \left[\frac{2t\cos(n\omega_0 t)}{4\pi^2 n^2} + \frac{n^2 4\pi^2 - 2}{8\pi^3 n^3} sin(n\omega_0 t)\right]_{-T}^{T} \tag{1.8}$$

Evaluating this integral:

$$a_n = 4 \cdot \left(\frac{4 \cdot (-1)^n}{4\pi^2 n^2} \right) = \frac{4 \cdot (-1)^n}{\pi^2 n^2} \tag{1.9}$$

For a_0, the integral is straightforward:

$$a_0 = \frac{1}{T} \int_0^T t^2 dt = \left[\frac{t^3}{3} \right]_0^T = \frac{1}{3} \tag{1.10}$$

So the first five coefficients are:

$$a_n = \left\{ \frac{-4}{\pi^2}, \frac{4}{16\pi^2}, \frac{-4}{36\pi^2}, \frac{4}{64\pi^2}, \frac{-4}{100\pi^2} \right\} \tag{1.11}$$

2. Use MATLAB to plot the original function and its fourier series up to the fifth harmonic. Put both graphs into the same diagram.

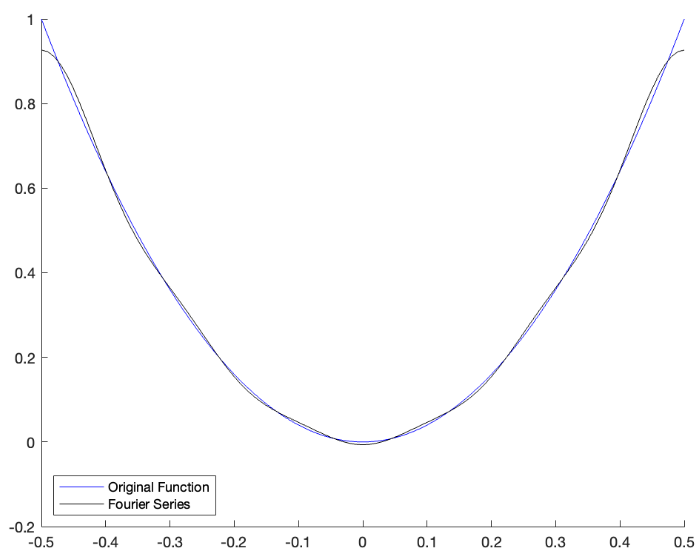

Figure 1.1: The plot of the original function, and its fourier series leading up to the fifth harmonic

1.3 Problem 3: FFT of a Rectangular Wave

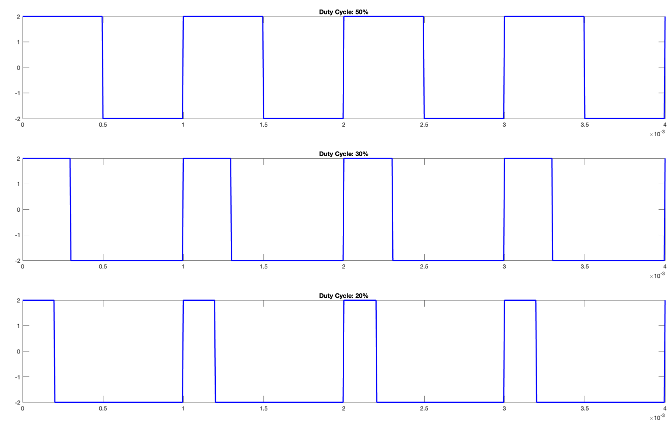

Figure 1.2: A figure showing the different rectangular waves based on their duty cycles. Plotted using MATLAB.

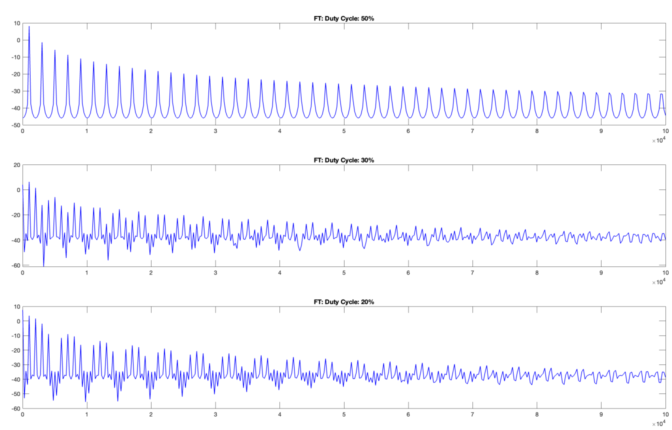

Figure 1.3: A figure showing the different rectangular waves' spectra based on their duty cycles. Plotted using MATLAB.

Following by the equation:

$$c_k = \frac{1}{k\pi}\sin(k\omega_0 T_1) \tag{1.12}$$

It is observed that for a lower T_1, for each frequency component the amplitude of the component is lower. This is due to the fact that the pulse width is lower, and thus the signal is more spread out in the time domain. This is also observed in the frequency domain, where the frequency components are more spread out, thereby more frequency components are required to represent the signal.

1.4 Problem 4: FFT of an audio file

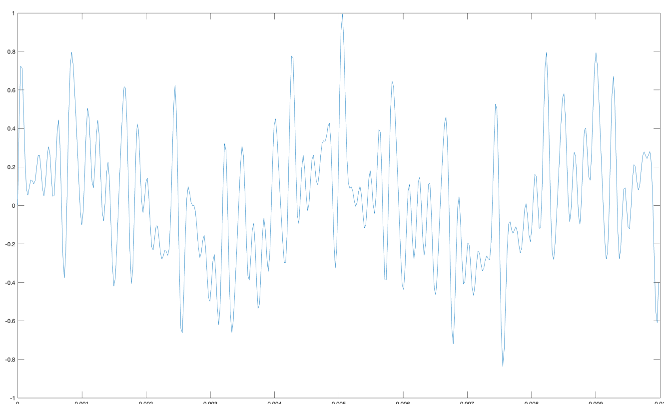

Figure 1.4: A figure showing the plot of the audio file

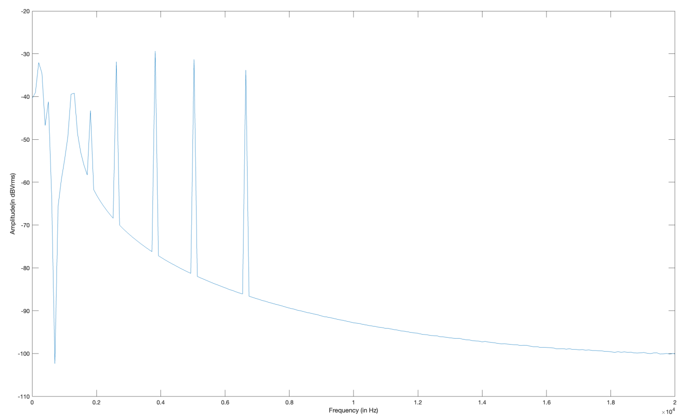

Figure 1.5: A figure showing the frequency spectrum of the audio file

The following tones are observed:

- G3 - 201Hz
- B4 - 503Hz
- E6 - 1308Hz
- A6 - 1812Hz
- E7 - 2617Hz
- A#7 - 3826Hz
- D#8 - 5034Hz
- G#8 - 6645Hz

The MATLAB code for all of the prelab problems are provided in the appendix.

2. Introduction

In this lab, the Fourier Transform was explored by means of the use of the FFT button on the oscilloscope. The Fourier Transform is a mathematical tool that allows us to convert a signal from the time domain to the frequency domain, thereby allowing the extraction of the frequency components that make up a signal.

A continuous time signal can be described by a sum of sinusoids of different frequencies, amplitudes, and phases. The Fourier Series is a mathematical tool that allows us to decompose a periodic signal into a sum of sinusoids, to use it, however, it must first be established what it means for a signal to be periodic. We say that a signal is periodic if for some positive period T the following holds:

$$x(t) = x(t + nT) \tag{2.1}$$

This must hold for all t. The fundamental period is the smallest positive period T for which the above holds. The fundamental frequency is defined as:

$$\omega_0 = \frac{2\pi}{T} \tag{2.2}$$

To determine the complex Fourier Series coefficients, the following equation is used,

$$c_\nu = \frac{1}{T}\int_T f(t)e^{-j\nu\omega_0 t}dt \tag{2.3}$$

Where $f(t)$ can then be expressed as,

$$f(t) = \sum_{\nu=-\infty}^{+\infty} c_\nu e^{j\nu\omega_0 t} \tag{2.4}$$

Where the DC component is obtained by substituting $\nu = 0$.

$$c_0 = \frac{1}{T}\int_T f(t)dt \tag{2.5}$$

The signal does not have to be represented by complex exponentials, indeed, it can also be represented by sines and cosines, where the fourier coefficients a_0, a_ν, and b_ν are used.

$$f(t) = \frac{a_0}{2} + \sum_{\nu=1}^{+\infty} a_\nu \cos(\nu\omega_0 t) + b_\nu \sin(\nu\omega_0 t) \tag{2.6}$$

2.1 The Square Wave

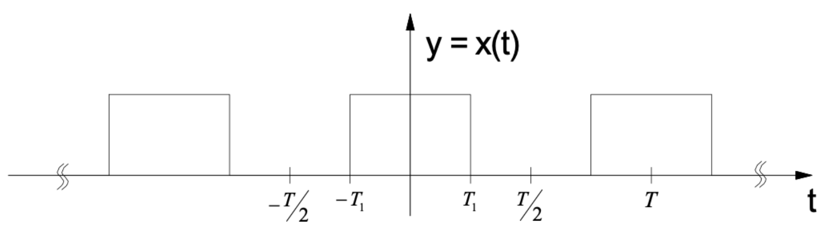

Figure 2.1: Square Wave

The square wave shown is a periodic signal that is defined as follows:

$$x(t) = \begin{cases} 1 & \text{if } |t| \le T/2 \\ 0 & \text{if } T_1 < |t| < T/2 \end{cases} \tag{2.7}$$

Using the equations for the Fourier Series coefficients, and knowing the signal is 1 between $-T/4$ and $T/4$, and 0 elsewhere,

$$c_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)dt = \frac{1}{T}\int_{-T/4}^{T/4} 1dt = \frac{1}{T}\left(\frac{T}{4} - \left(-\frac{T}{4}\right)\right) = \frac{1}{2} \tag{2.8}$$

For the c_ν coefficients,

$$\begin{aligned} c_\nu &= \frac{1}{T}\int_{-T_1}^{T_1} x(t)e^{-j\nu\omega_0 t}dt \\ &= -\frac{1}{j\nu\omega_0 T} e^{-j\nu\omega_0 t}\Big|_{-T_1}^{T_1} \\ &= \frac{2}{\nu\omega_0 T}\left(\frac{e^{j\nu\omega_0 T_1} - e^{-j\nu\omega_0 T_1}}{2j}\right) \end{aligned} \tag{2.9}$$

Where from Euler's identities, it can be extracted:

$$c_\nu = \frac{1}{\nu\pi}\sin(\nu\omega_0 T_1) \tag{2.10}$$

Where T is the period, and T_1 is the width of the pulses. Similarly, for the a_ν and b_ν coefficients extracted from the complex exponential form of the Fourier Series,

$$a_\nu = 2\Re(c_\nu) = \frac{2}{\nu\pi}\sin\left(\nu\frac{\pi}{2}\right) = \begin{cases} 0 & \nu \text{ even} \\ \frac{2}{\nu\pi} & \nu \text{ odd} \end{cases} \tag{2.11}$$

$$b_\nu = -2\Im(c_\nu) = 0 \tag{2.12}$$

Which leads to,

$$x(t) = \frac{1}{2} + \frac{2}{\pi}\sum_{\nu=1,3,5,\ldots}^{\infty} sin(\nu\frac{\pi}{2}) \cdot \frac{cos(\nu\omega_0 t)}{\nu} \tag{2.13}$$

The Gibbs Phenomenon is the overshoot of the Fourier Series approximation near the discontinuities of a periodic signal, which most surely can be observed in the case of the square wave. The discontinuities in the pulses makes it necessary to have more and more sinusoids to approximate it, and even then, the approximation is not perfect, leading to ”ringing” near the discontinuities.

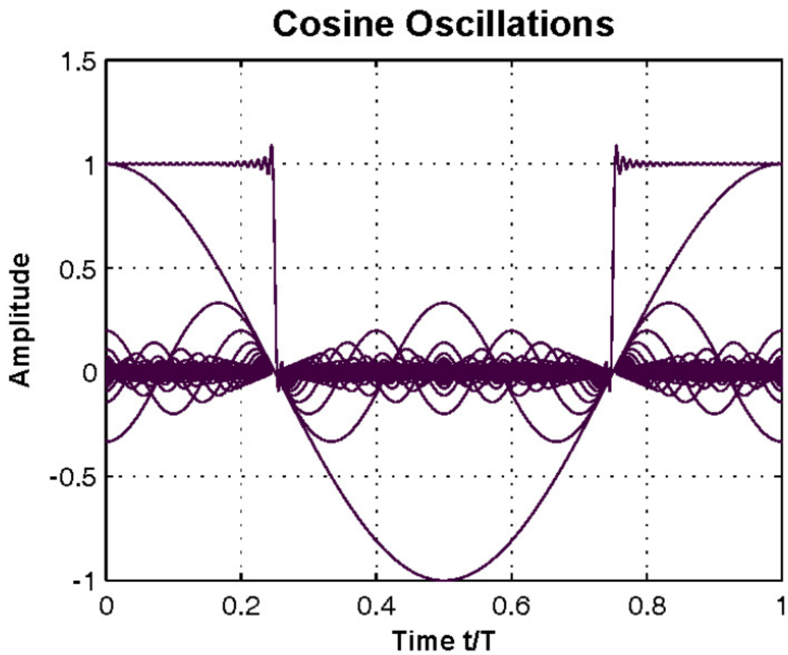

Figure 2.2: Gibbs Phenomenon, the overshoot near the discontinuities of the square wave approximated by 50 sinusoids.

2.2 The Fourier Transform

The Continuous Time Fourier Transform is what is obtained when the period of the Fourier Series goes to infinity. It is defined by,

$$X(j\omega) = \int_{\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{2.14}$$

And the inverse is given by,

$$x(t) = \frac{1}{2\pi}\int_{\infty}^{\infty} x(t)e^{j\omega t}d\omega \tag{2.15}$$

For a square pulse, which is essentially a square wave of infinite period, the Fourier Transform is given by,

$$X(j\omega) = \int_{-\tau/2}^{\tau/2} e^{-j\omega t}dt \tag{2.16}$$

By using Euler's trigonometric identities, it is obtained that the Fourier Transform of the square pulse is given by,

$$X(j\omega) = \tau \cdot \left[\frac{\sin(\frac{\omega\tau}{2})}{\frac{\omega\tau}{2}}\right] = \tau \cdot sinc\left(\frac{\omega\tau}{2}\right) \tag{2.17}$$

2.3 The Discrete Fourier Series

The Discrete Fourier Series is the discrete time equivalent of the Fourier Series, and the coefficients are defined as follows:

$$a_\nu = \frac{1}{N} \sum_{n=-N_1}^{N_1} e^{-j\frac{2\pi\nu n}{N}} \tag{2.18}$$

Furthermore, the fourier series summation can be expressed as,

$$x[n] = \frac{1}{N} \sum_{\nu=(N)} a_\nu e^{j\frac{2\pi\nu n}{N}} \tag{2.19}$$

2.4 The Discrete Fourier Transform

Because computers can not handle continuous time signals, an analogous transform to the Fourier Transform is necessary, one that can handle discrete time signals. This is the Discrete Fourier Transform, which is defined as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}} \tag{2.20}$$

And the inverse is given by,

$$x[k] = \frac{1}{N} \sum_{n=0}^{N-1} X[n]e^{j\frac{2\pi kn}{N}} \tag{2.21}$$

The DFT, and the more commonly used algorithm to perform the DFT, the FFT, are widely used for signal filtering, for correlation analysis and for spectral analysis. The FFT simply uses the properties of periodicity, symmetry, and uses a divide-and-conquer approach to reduce the number of computations needed to perform the DFT. It is not an approximation of the DFT, it is simply a more efficient way to compute it.

During the experiment, the Hanning Window is used on the oscilloscope as it is the best window to use when the signal is periodic. A window function simply cuts out the signal so that the FFT can be performed on it. The Hanning Window is some sort of rectangular pulse multiplied by the time signal, however, which leads to a sinc in the frequency domain. It will affect the transform, but it is the only viable option to perform the FFT on the signal.

3. Execution

3.1 Problem 1: FFT of Single Tone sinusoidal wave

1. The function generator was set to a sinusoidal wave with a 500Hz frequency, $2V_{pp}$ amplitude, and no offset. The measure function was used to verify all the properties of the signal, and a hardcopy was subsequently taken.

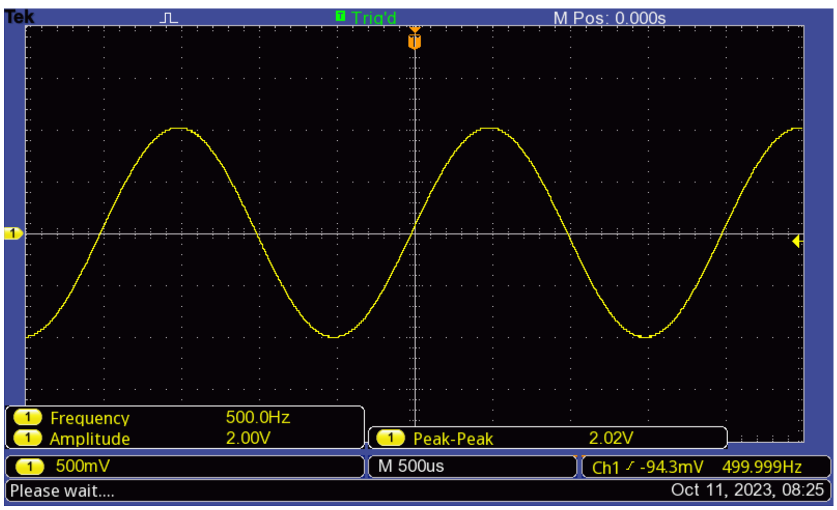

Figure 3.1: Hardcopy of the signal

2. Afterward, the FFT spectrum was obtained via the spectrum's FFT function. The cursor was used to measure the properties of the spectrum, and then a hardcopy was taken. Furthermore, another hardcopy was taken of the zoomed in spectrum.

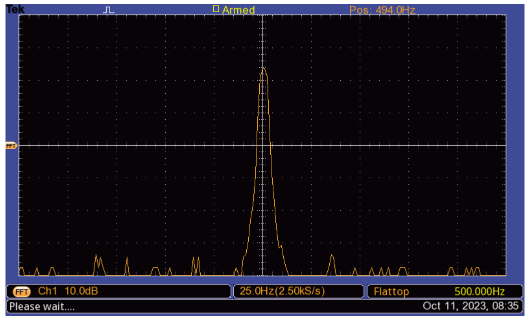

Figure 3.2: Hardcopy of the FFT spectrum, showing 494Hz.

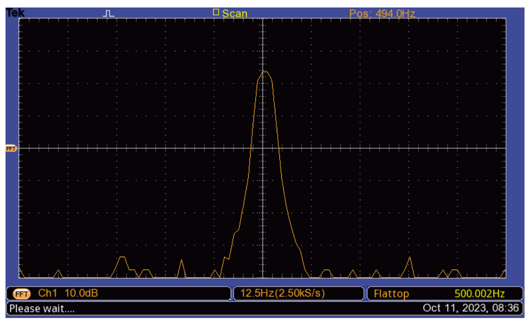

Figure 3.3: Hardcopy of the zoomed in FFT spectrum, showing 494Hz.

What is immediately noticed, however, is that the position seems to show 494Hz. This is in fact an error of 1%, when the usual error should be around 0.1%. The reason this error shows up may be due to the miscalibration of the Oscilloscope.

3. For the sinusoidal wave with a 2KHz frequency(without any offset) to have a 0dB spectrum peak, the V_{pp} was found to be around $2.7V_{pp}$. A hardcopy of the time-domain signal was taken.

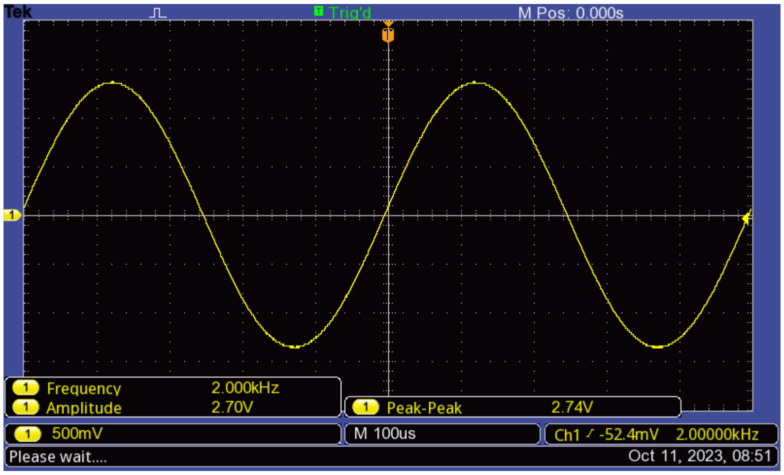

Figure 3.4: Hardcopy of the time-domain signal with 0dB spectrum peak.

Then, a hardcopy is taken of the frequency spectrum that is obtained by using the FFT function on the oscilloscope.

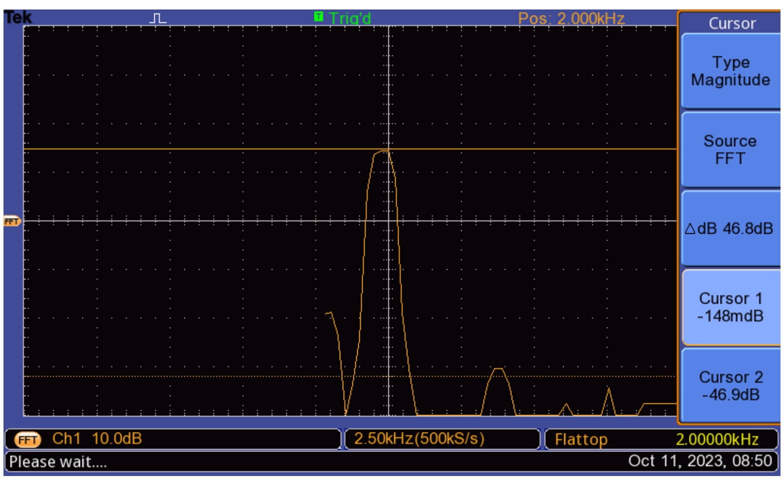

Figure 3.5: Hardcopy of the frequency spectrum with 0dB spectrum peak.

It is observed that the cursor shows a dB of −148mdB which is close to 0dB. The small error is because of the Oscilloscope's resolution. It also shows the 2KHz frequency.

3.2 Problem 2:

1. The function generator is used to generate a square wave at $2V_{pp}$ amplitude, a 1ms period, and no offset. A hardcopy of the signal is then taken.

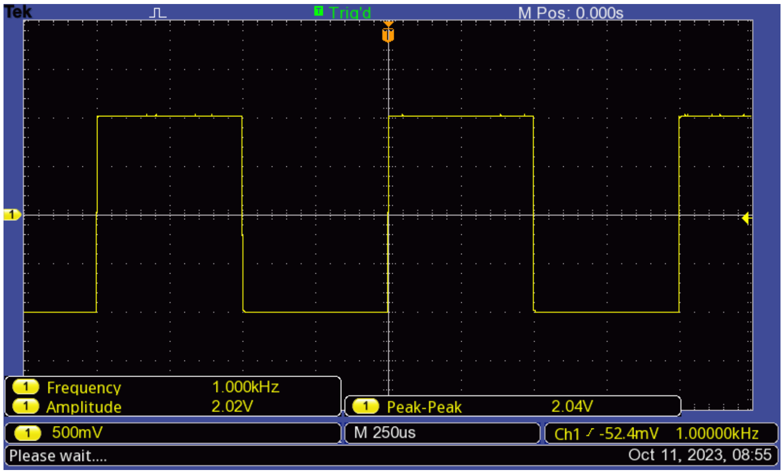

Figure 3.6: Hardcopy of the square wave signal.

2. The FFT spectrum is then obtained, and zoomed into, subsequently hardcopies of each frequency component are taken. The harmonics observed are as follows:

   - 1050Hz: -989mdB
   - 2950Hz: -10.1dB
   - 5000Hz: -14.9dB
   - 7000Hz: -17.7dB

   Hardcopies are taken of the cursor over each of the harmonics.

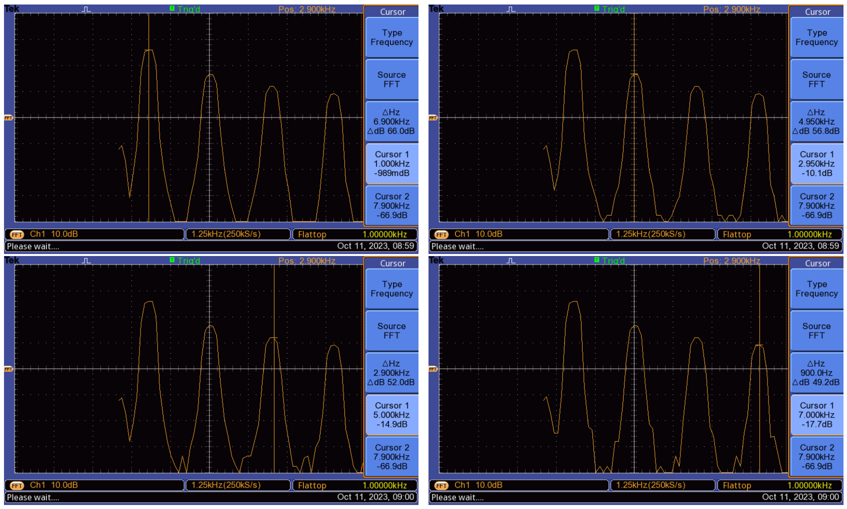

Figure 3.7: The harmonics of the square wave.

Which are pretty consistent with theory and the square wave on the screen, in that the harmonics are odd multiples of the fundamental frequency.

3. The FFT spectrum is then obtained for 20% duty cycles, the amplitudes of the first four harmonics are measured and hardcopies are taken of the signal both in time and frequency domain.

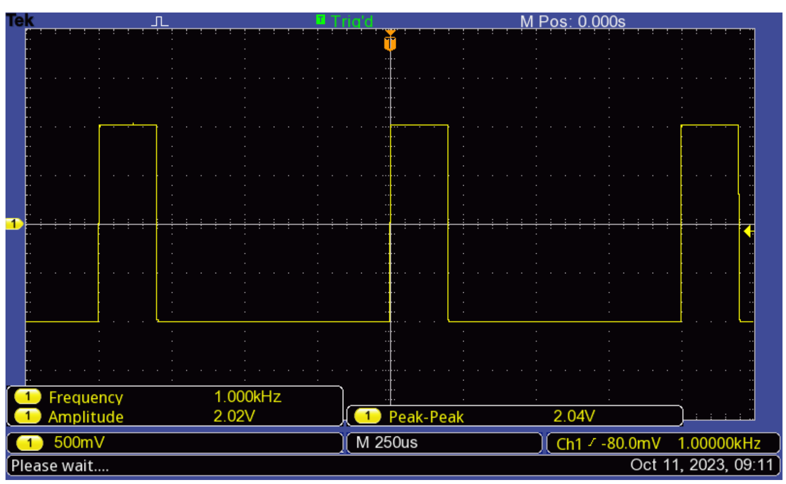

Figure 3.8: Hardcopy of the 20% duty cycle square wave.

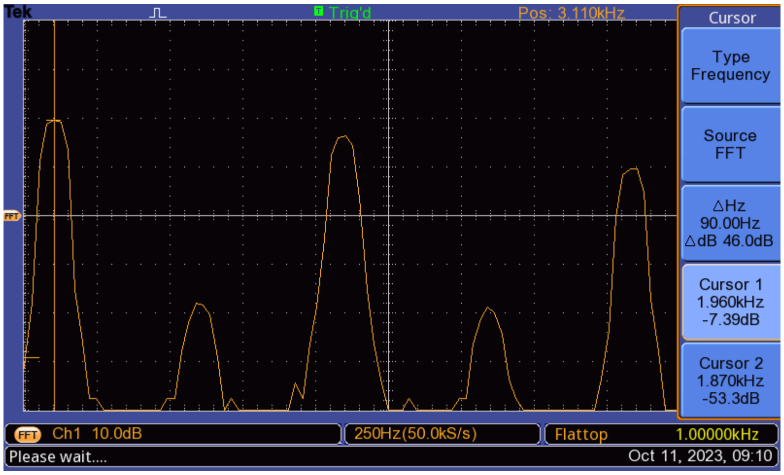

Figure 3.9: Hardcopy of the 20% duty cycle square wave FFT spectrum.

The harmonics were then taken by using the cursor, the following values were obtained for the non-DC harmonics:

- 1010Hz: -5.39dB
- 2010Hz: -7.39dB
- 2970Hz: -10.9dB
- 3980Hz: -17.3dB

Because the signal is at a 20% duty cycle, the harmonics are no longer odd multiples of the fundamental frequency.

3.3 Problem 3: FFT of Multiple-Tone sinusoidal wave

The following circuit is assembled.

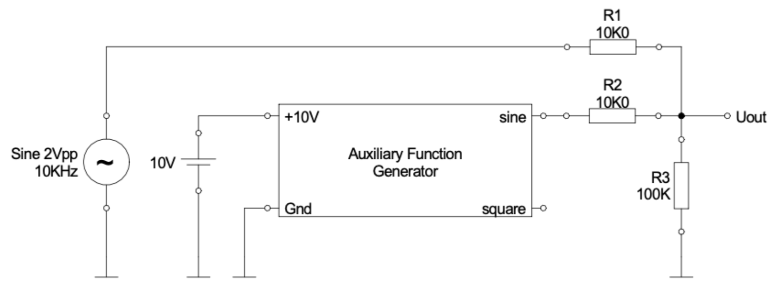

Figure 3.10: The circuit used for problem 3.

Hardcopies of the signal and the signal's spectrum are then taken.

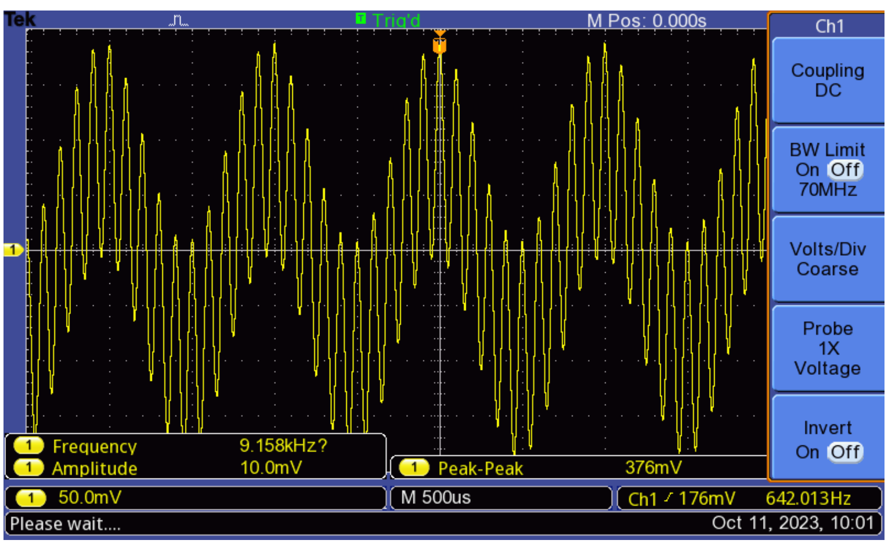

Figure 3.11: Hardcopy of the signal.

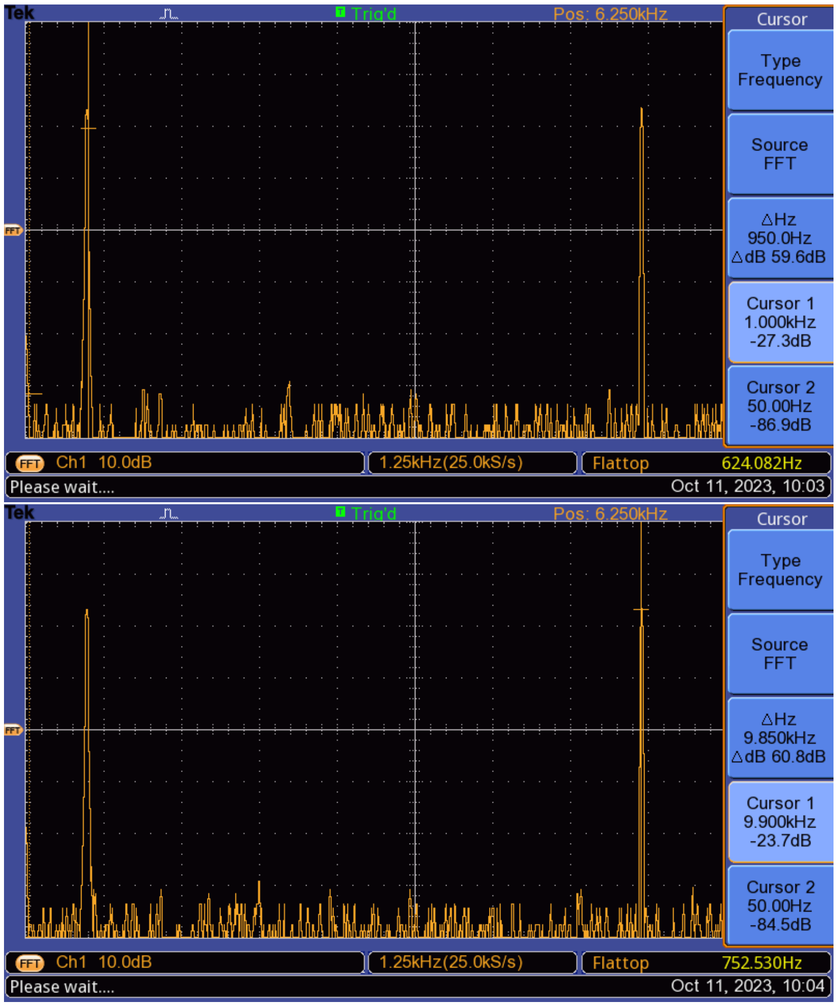

Figure 3.12: Hardcopy of the signal's spectrum, and the frequencies measured at 1000kHz and 9900kHz, respectively

4. Evaluation

4.1 Problem 1: FFT of Single Tone sinusoidal wave

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

   The reference value of the oscilloscope for 0dB is $1V_{\text{rms}}$ and $1.4V_{pp}$.

2. The MATLAB calculation of the FFT of the signal is shown in the figure below.

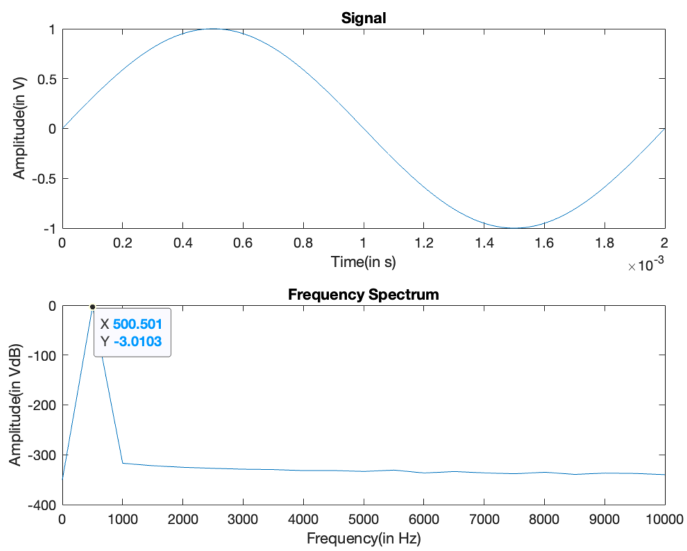

Figure 4.1: FFT of Single Tone sinusoidal wave

The calculated spectra seems to be in-line with what is seen in the oscilloscope, although there are no minor disturbances which are caused by the leakage of the windowing function used by the oscilloscope. The code used to generate the FFT is shown in the appendix.

3. The MATLAB calculation of the FFT of the signal is shown in the figure below. In order to get a 0dB spectral peak, the voltage must be $sqrt(2)$, which is 1.414V or $2.82V_{pp}$.

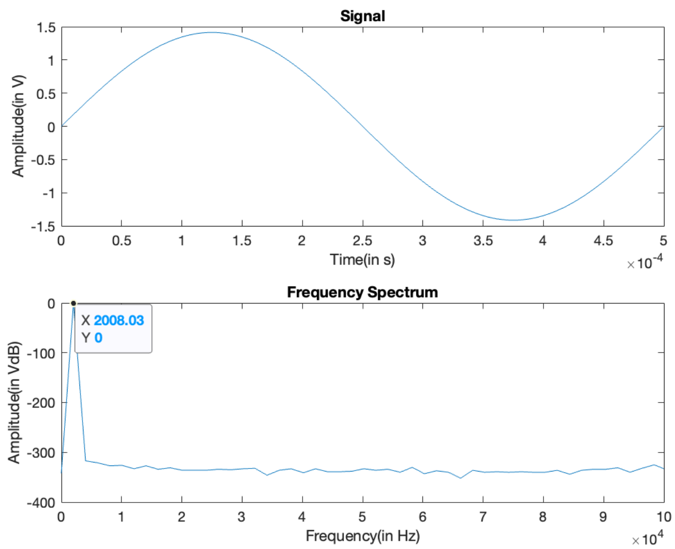

Figure 4.2: FFT showing 0dB spectral peak

The code used to generate the FFT is shown in the appendix.

4. The results are in line with what is observed in the oscilloscope. The oscilloscope shows a -148mdB spectral peak when the voltage is $2.7V_{pp}$, which is close to the theoretical value of 0dB at $2.82V_{pp}$, the reason it is not exactly 0dB is because of the resolution of the oscilloscope.

4.2 Problem 2: FFT of a Square Wave

1. When the frequency scale is expanded to measure the frequency components of the signal, the bandwidth will decrease. The higher the bandwidth, the lower the resolution of the oscilloscope.

2. In the 20% duty cycle, it is observed that the signal can no longer be characterized by simply odd and even harmonics. Furthermore, every fifth harmonic has a significantly lower amplitude.

4.3 Problem 3: FFT of a multiple-tone signal

1. It is observed that when adding two signals together(a 1000KHz and 10,000KHz signal), the FFT of the signal is the sum of the FFT of the two signals. This is because the FFT is a linear operation, and we see the same behaviour in the oscilloscope, where it is observed that there are spikes at 1000KHz and 9900KHz, and knowing that both signals are sinusoids, it can be concluded that the resulting FFT is the sum of the FFT of the two signals.

5. Conclusion

In this experiment properties of the FFT function on the oscilloscope were observed. In general, the FFT of the oscilloscope is accurate, however the windowing function which causes leakage is not taken into account, which causes minor disturbances in the frequency spectrum.

The oscilloscope may also not have been calibrated properly, as there was a 1% error in the frequency, which is unusual as the oscilloscope should usually be at around 0.01% error.

Furthermore, the oscilloscope has a limited resolution, which causes the spectral peak to be slightly off from the theoretical value of 0dB when the voltage was $2.7V_{pp}$, where the oscilloscope would fluctuate between -148mdB to 200mdB, no matter the adjustment made.

The oscilloscope also has a limited bandwidth, which causes the resolution to decrease when the frequency scale is expanded. Furthermore, when the duty cycle is decreased to 20%, the signal can no longer be characterized by simply odd and even harmonics, and every fifth harmonic has a significantly lower amplitude. Finally, when adding two signals together, the FFT of the signal is the sum of the FFT of the two signals, which is expected since the FFT is a linear operation.

6. References

1. Oscilloscope Manual
2. Lab Manual for Signals and Systems
3. MATLAB Documentation

7. Appendix

7.1 Prelab Problem Code

```
% Prelab:

%% Problem 2:
t = -0.5:0.01:0.5;
f = 4*t.^2;
f_fs = 1/3;

% Fourier series representation of the function
for i=1:5
    % w_0 = 2pi
    % T = 1
    a_i = (4*(-1)^i)/(pi^2*i^2);
    f_fs = f_fs + (a_i .* cos(i.*2.*pi.*t));
end
hold on

plot(t, f, "blue");
plot(t, f_fs, "black");
xlim([-0.5, 0.5]);

legend({"Original Function", ...
    "Fourier Series" ...
    }, "Location", "southwest")

%% Problem 3:

%% Time plots

[t_fifty, signal_fifty, f_fifty, y_fifty] = get_square(50);
[t_thirty, signal_thirty, f_thirty, y_thirty] = get_square(30);
[t_twenty, signal_twenty, f_twenty, y_twenty] = get_square(20);
```

```
subplot(3, 1, 1);
plot(t_fifty, signal_fifty, "blue", "LineWidth", 2);
title("Duty Cycle: 50%");

subplot(3, 1, 2);
plot(t_thirty, signal_thirty, "blue", "LineWidth", 2);
title("Duty Cycle: 30%");

subplot(3, 1, 3);
plot(t_twenty, signal_twenty, "blue", "LineWidth", 2);
title("Duty Cycle: 20%");

%% Frequency plots
subplot(3, 1, 1);
plot(f_fifty, y_fifty, "blue", "LineWidth", 1);
title("FT: Duty Cycle: 50%");


subplot(3, 1, 2);
plot(f_thirty, y_thirty, "blue", "LineWidth", 1);
title("FT: Duty Cycle: 30%");


subplot(3, 1, 3);
plot(f_twenty, y_twenty, "blue", "LineWidth", 1);
title("FT: Duty Cycle: 20%");


%% Problem 4:

[y, Fs] = audioread("s_samp.wav");

N_samples = Fs * 10E-3;
y_first = y(1:N_samples);
t = 0:1/Fs:((10E-3)-1/Fs);
plot(t, y_first);
%%
```

```
N_samples = length(y);
Y = fft(y);
rms = sqrt(mean(abs(Y.^2)));

Fs_nyquist = Fs / 2;

Y_single = 2 * abs(Y) / N_samples;
Y_single = Y_single(1:floor(N_samples/2));
f = linspace(0,Fs_nyquist,length(Y_single));

Y_db = 20*log10(Y_single ./ rms);
Y_db(Y_db == -Inf) = 0;

% plot(f, Y_single);

plot(f, Y_db);
ylabel("Amplitude(in dBVrms)")
xlabel("Frequency (in Hz)");

%% Functions

function [t, signal, f, y_single_db] = get_square(duty_cycle)
    period = 1e-3;
    Fs = 200e3;
    frequency = 1/period;
    Vpp = 2;
    duration = period * 6;

    % The signal
    t = 0:1/Fs:duration;
    signal = Vpp*square((2*pi*frequency)*t, duty_cycle);
    plot(t, signal, "blue", "LineWidth", 2);

    % The fourier transform of the signal
    rms_value = sqrt(mean(signal.^2));
    N = length(signal); % The length of the signal
```

```
    y = fft(signal, N);
    y_mag = 2*(abs(y)/N); % Magnitudes of y

    y_single = y_mag(1:floor(N/2)) * 2;
    f_nyquist = Fs / 2;

    y_single_db = 20*log10(y_single / rms_value);
    f = linspace(0, f_nyquist, length(y_single));
end
```

7.2 Code for the evaluation problems:

```
%% Problem 2:

F = 500;
T = 1/F;
Fs = 1E6;
Ts = 1/Fs;
w = 2*pi*F;

Vpp = 2;
Vamp = Vpp / 2;
t = 0:Ts:T - Ts;
signal = Vamp * sin(w*t);

subplot(2, 1, 1);
plot(t, signal);
title("Signal");
ylabel("Amplitude(in V)");
xlabel("Time(in s)")

N = length(signal);

y = fft(signal);
y = 2 * (abs(y) / N);
y = y(1:floor(N/2));
```

```
f = linspace(0, Fs/2, length(y));
subplot(2, 1, 2);
plot(f, mag2db(y / sqrt(2)));
title("Frequency Spectrum");

xlim([-0, 1e4]);
ylabel("Amplitude(in VdB)");
xlabel("Frequency(in Hz)")

%% Problem 3
close all
clear

F = 2000;
T = 1/F;
Fs = 1E6;
Ts = 1/Fs;
w = 2*pi*F;

t = 0:Ts:T - Ts;
V_amp = sqrt(2);
signal = V_amp*sin(w*t);

subplot(2, 1, 1);
plot(t, signal);
title("Signal");
ylabel("Amplitude(in V)");
xlabel("Time(in s)")

N = length(signal);

y = fft(signal);
y = 2 * (abs(y) / N);
y = y(1:floor(N/2));

f = linspace(0, Fs/2, length(y));
subplot(2, 1, 2);
plot(f, floor(mag2db(y / sqrt(2))));
```

```
title("Frequency Spectrum");

xlim([-0, 1e5]);
ylabel("Amplitude(in VdB)");
xlabel("Frequency(in Hz)")
```

7.3 Prelab for Lab 4

7.3.1 Problem 1: The Sampling Theorem

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

   The low-pass filter is used to remove noise, and to remove frequency components that may be higher than the Nyquist frequency. This is necessary to prevent not accounting for the higher frequency components, which would result in aliasing.

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

   The minimum sampling frequency for a pure sine wave input at 3KHz would be $f_s > 2 * f_{Nyquist} = 6\text{KHz}$

3. **What is the Nyquist frequency?**

   The Nyquist frequency is the highest frequency that can be represented in a sampled signal. It is half the sampling frequency and defined as $f_{Nyquist} = \frac{f_s}{2}$

4. **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 frequencies would be:

   (a) 500Hz: 500Hz

   (b) 2.5Khz: 0KHz (DC component)

   (c) 5KHz: 0KHz (Aliased)

(d) 5.5KHz: 0.5KHz (Aliased, by $f - f_{sampling}$)

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

   The frequencies would be: 37Hz, 67Hz, 97Hz.

7.3.2 Problem 2: Impulse Train Sampling and Real Sampling

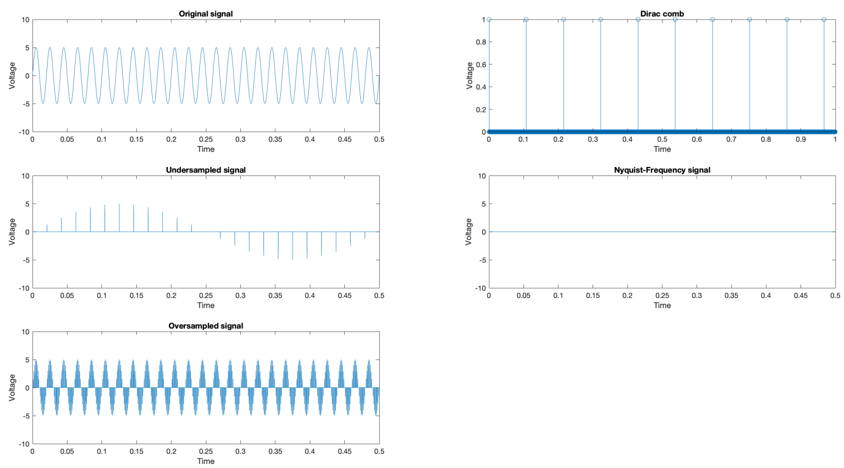

Figure 7.1: Impulse train sampling

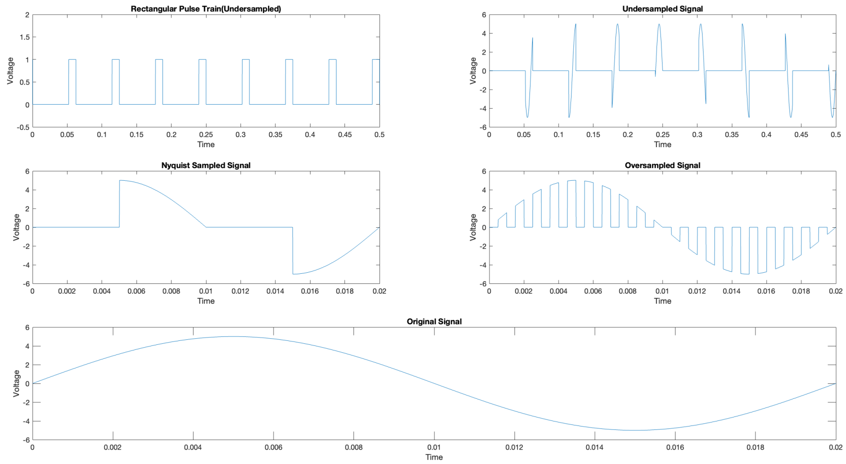

Figure 7.2: Rectangular Pulse Train Sampling

The code used to perform the sampling is given below.

```
%% Prelab:

% first part:

f = 50; % frequency, in hertz
f_n = 2*f; % nyquist frequency, in hertz

samples=1E5;
t = 0:1/samples:1; % 100kS

V_pp = 5;
w = 2*pi*f;
x = V_pp * sin(w .* t);

% Original signal:
subplot(3, 2, 1);
plot(t, x);
ylabel("Voltage")
xlabel("Time")
title("Original signal");
```

```
ylim([-10, 10]);
xlim([0, 0.5]);

% Dirac comb
subplot(3, 2, 2);
p = zeros(size(t));
p(1:10750:length(t)) = 1;
stem(t, p);
ylabel("Voltage")
xlabel("Time")
title("Dirac comb");
ylim([0, 1]);
xlim([0, 1]);

% Undersampled signal:
fs = 48; % In hertz

p = zeros(size(t));
p(1:samples/fs:length(t)) = 1;

subplot(3, 2, 3);
plot(t, p .* x);
ylabel("Voltage")
xlabel("Time")
title("Undersampled signal");
ylim([-10, 10]);
xlim([0, 0.5]);

% Nyquist-Frequency sampled signal
fs = 100; % In hertz

p = zeros(size(t));
p(1:samples/fs:length(t)) = 1;

subplot(3, 2, 4);
plot(t, p .* x);
ylabel("Voltage")
xlabel("Time")
```

```
title("Nyquist-Frequency signal");

ylim([-10, 10]);
xlim([0, 0.5]);

% Oversampled signal
fs = 1000; % In hertz

p = zeros(size(t));
p(1:samples/fs:length(t)) = 1;

subplot(3, 2, 5);
plot(t, p .* x);
ylabel("Voltage")
xlabel("Time")
title("Oversampled signal");

ylim([-10, 10]);
xlim([0, 0.5]);

%%

clear
close all

f = 50;
t = 0:1E-5:1;
V_pp = 5;
w = 2*pi*f;
x = V_pp * sin(w .* t);

% Underdamped signal:
fs = 48; % In hertz
T = 1/fs;
T0 = 0.5 * 1/fs;
p = gen_pulse_train(t, T0, T);

subplot(3, 2, 1);
```

```
plot(t, p);
title("Rectangular Pulse Train(Undersampled)");
ylabel("Voltage")
xlabel("Time")
xlim([0, 1/2]);
ylim([-0.5, 2]);

subplot(3, 2, 2);
plot(t, p .* x);
ylabel("Voltage")
xlabel("Time")
title("Undersampled Signal")
xlim([0, 1/2]);
ylim([-6, 6]);


% Nyquist Sampled signal:
fs = 100; % In hertz
T = 1/fs;
T0 = 0.5 * 1/fs;
p = gen_pulse_train(t, T0, T);

subplot(3, 2, 3);
plot(t, p .* x);
ylabel("Voltage")
xlabel("Time")
title("Nyquist Sampled Signal")
xlim([0, 1/50]);
ylim([-6, 6]);

% Overly Sampled signal:
fs = 1000; % In hertz
T = 1/fs;
T0 = 0.5 * 1/fs;
p = gen_pulse_train(t, T0, T);

subplot(3, 2, 4);
plot(t, p .* x);
```

```
ylabel("Voltage")
xlabel("Time")
title("Oversampled Signal")
xlim([0, 1/50]);
ylim([-6, 6]);

subplot(3, 2, [5 6]);
plot(t, x);
ylabel("Voltage")
xlabel("Time")
title("Original Signal")
xlim([0, 1/50]);
ylim([-6, 6]);

% Function to generate pulse train
function [p] = gen_pulse_train(t, T0, T)
      p = zeros(size(t));
      for k=1:length(t)
            if mod(t(k), T) == 0
            left = t(k) - T0;
            right = t(k) + T0;
            p((t > left) & (t < right)) = 1;
            else
            p(k) = 0;
            end
      end
end
```

7.3.3 Problem 3: Sampling using a Sampling bridge

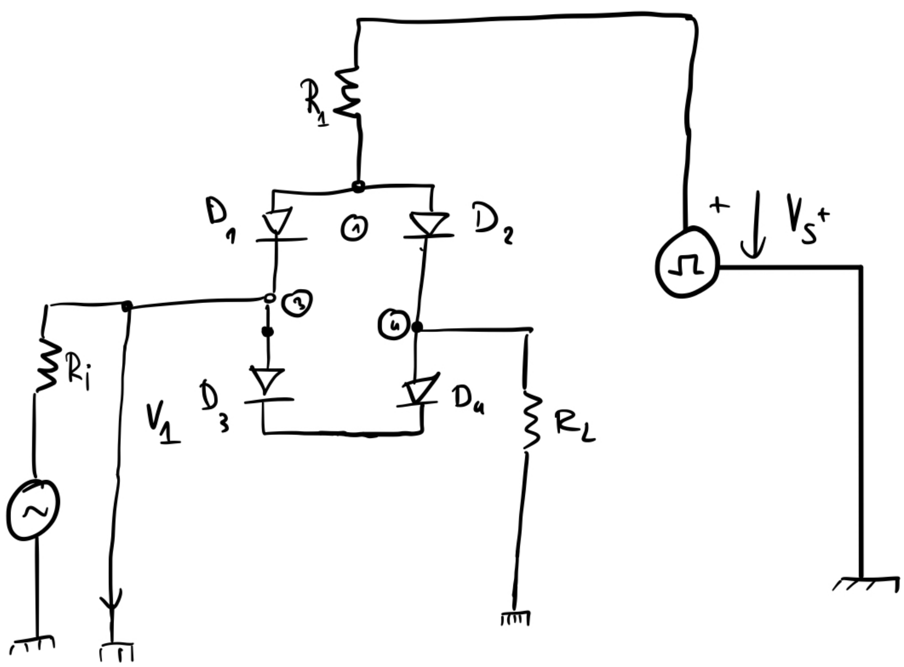

Figure 7.3: The modified Sampling Bridge circuit.

The reason this sampling bridge circuit works is due to the polarity of the two different voltage sources given in the original circuit being practically the same, with the potential being the same for both sources, we can simply omit one and lead current into the same square wave generator.

7.4 Hardcopies for Lab 4