Experiment 3

Fourier Synthesis of Periodic Waveforms

Report

Important: Marking of all coursework is anonymous. Do not include your name, student ID number, group number, email or any other personal information in your report or in the name of the file submitted online. A penalty will be applied to submissions that do not meet this requirement.

Part A (30 Marks)

A.1) Provide the Matlab code required to synthesise the waveform in equation 12 and the resulting waveform. [5 marks]

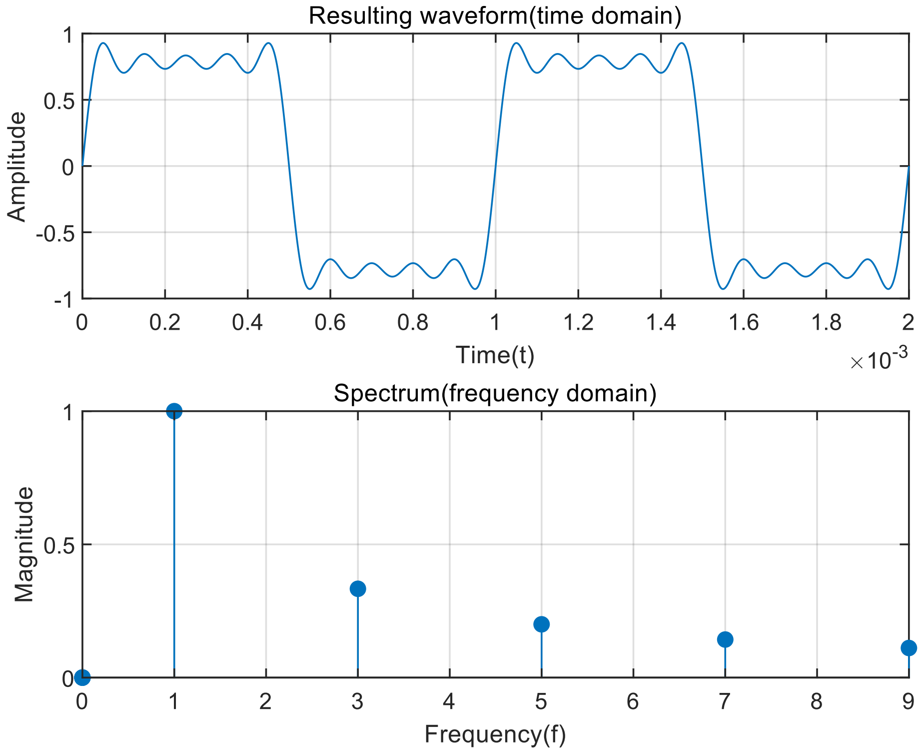
MATLAB code:

```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V0 = 1;
t = linspace(0,2*T,10000);%Define time domain range and step size
harmonics =10;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if mod(n,2)
        fn = V0*sin(n*2*pi*f0*t)*(1/n); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V0/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);
plot(t,ft);%Draw time domain waveform
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2);
stem(f,An,"filled");%Draw frequency domain
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Resulting waveform:



A.2) Explain the features of the resulting waveform (peak-to-peak amplitude, symmetry, ripple, etc.) [5 marks]

1. Peak-to-peak amplitude

As can be seen from the graph, the waveform has a maximum value of 0.9286 V and a minimum value of -0.9286 V during one cycle. Thus, the final inter-peak voltage is **1.8572V**. The voltage is still somewhat different from the theoretical value. Assuming that V_0 is 1V and that the harmonics are close to infinity, the amplitude between the peaks of the theoretical square wave may be close to **1.57V** (at harmonics = 100000). When only the first 10 harmonics are considered, some errors occur due to the Gibbs phenomenon.

2. Symmetry

It is easy to see that the waveform has symmetry. When f0 is 1000 Hz, the waveform is centrosymmetric about x = 0.5, 1, 1.5... (+k*0.5) Centro symmetry. The waveform is also symmetric about 0.25, 0.75, 1.25... (+k*0.5) symmetry. By increasing the time domain, it can be seen that the waveform has an infinite number of symmetry axes. It is easy to see that this waveform is an odd waveform, and also that after shifting the waveform by half a cycle the waveform is the opposite of the original waveform so it is an odd harmonic.

3. Ripple

A series of ripples appears in each cycle of the waveform. In each region where ripples appear, the waveform will have the largest fluctuations at the start and end positions, and smaller fluctuations in the middle section, and the smaller fluctuations occupy most of the time. The waveform has nine sub-peaks and nine sub-valleys. When calculating a specific percentage of overshoot for a synthesised waveform, the following equation can be used: Percentage of overshoot = $\frac{A_{max}-A_{steady}}{A_{steady}} = \frac{1.8572-1.57}{1.57} \times 100\% \approx 18.3\%$

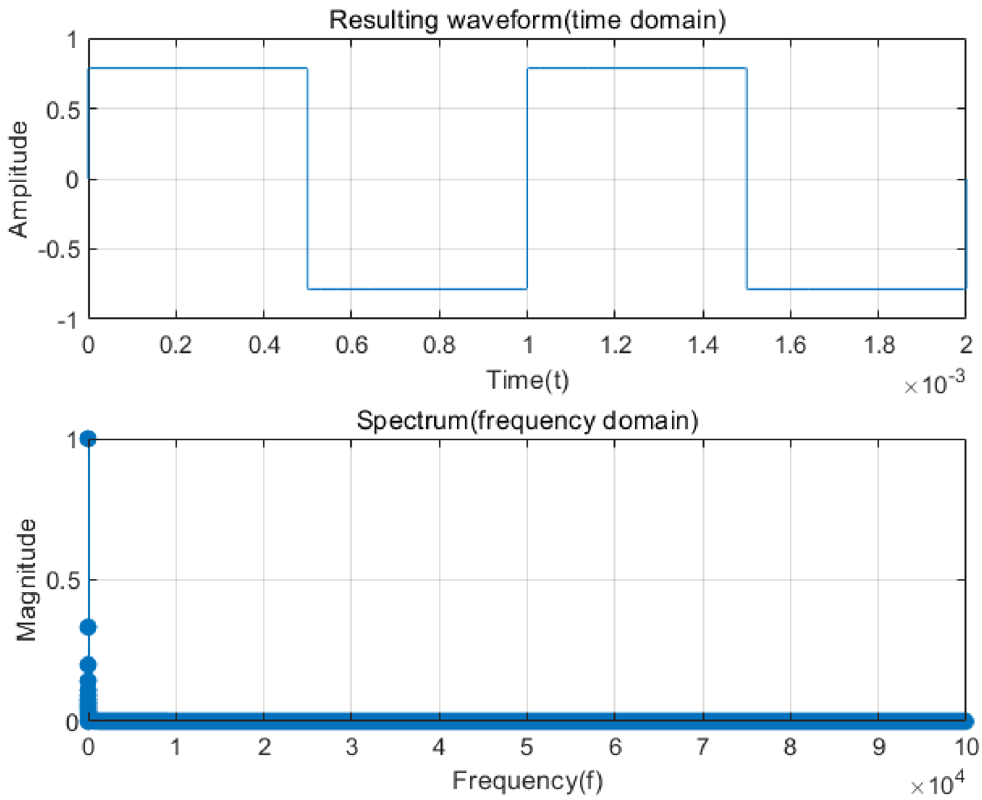
With this method, the overshoot percentage for this waveform will be close to 18.3% when harmonics = 10. If the harmonic values continue to increase, the overshoot percentage for this waveform will be closer to 0%.

4. Period and frequency domain

The base frequency f_0 is 1000Hz in the graph, so the period is 1ms. For ease of observation, two cycles of the waveform are shown. The spectrum of the waveform shows that this waveform is an odd harmonic with a0=0, which confirms the previous observations of the waveform.

A.3) How do you think the waveform would look if an unlimited number of harmonics was available (i.e., *n* goes to ∞)? To support your answer, provide a couple of figures along with their associated code. [5 marks]

If the number of harmonics is infinite, the waveform will become a square wave. The code has been designed with this in mind and is verified by directly changing the number of harmonics to a large value. When the number of harmonics is 100000 the waveform is already very close to a square wave. This is illustrated below:



A.4) Referring to Equation 3, what is the value of a_o? [5 marks]

Equation 3 is:

$$a_0 = \frac{1}{T}\int_0^T f(t)dt$$

Where *f(t)* is:

$$f(t) = V_0(\sin\omega t + \frac{1}{3}\sin 3\omega t + \frac{1}{5}\sin 5\omega t + \frac{1}{7}\sin 7\omega t + \frac{1}{9}\sin 9\omega t)$$

Hance:

$$a_0 = \frac{1}{0.001}\int_0^{0.001} f(t)dt = 0$$

Where V_0=1, T=0.001s, $w = 2\pi f_0$, f_0 = *1000Hz*, f_0 = 1/T.

a_0 is the average amplitude of the waveform, and since the image is a periodic function and the point of Centro symmetry is also on the x-axis, it is easy to deduce that a_0 is 0. In addition, the spectrum of the waveform also shows that the 0th harmonic component, which is the DC component, is 0, which also means that $a_0 = 0$.

A.5) Find an expression (in terms of *n*) for a_n and b_n. [5 marks]

We know that $a_0 = 0$, V_0=1, T=0.001s, $w = 2\pi f_0$, f_0 = *1000Hz*, f_0 = 1/T

So that,

$$f(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos n2\pi f_0 t + b_n \sin n2\pi f_0 t)$$

Where a_n and b_n as follow,

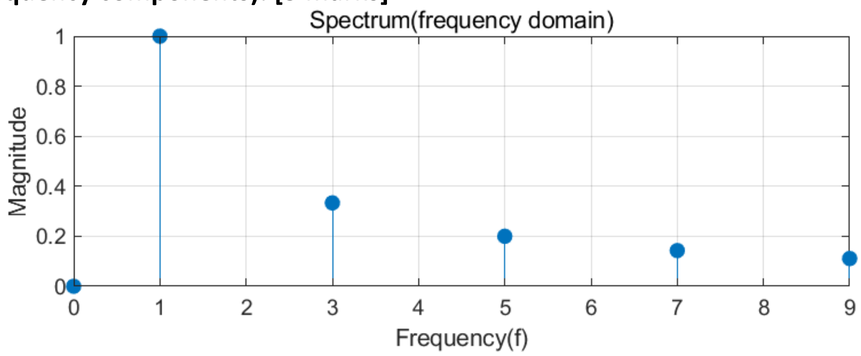
$$a_n = 2000\int_0^{0.001} f(t)\cdot cos\,(2000n\pi t)dt$$

$$b_n = 2000\int_0^{0.001} f(t)\cdot sin\,(2000n\pi t)dt$$

By observing the equation, an is the coefficient of the cosine term in the equation f(t). Since this is an odd wave, an will only be 0. Again, consider that bn is the coefficient of the sine term in the equation f(t). Thus,

$$a_n = 0 \text{ and } b_n = 1/n$$

A.6) Plot the spectrum (frequency domain view) of *f(t)* using $c_n = \sqrt{(a_n^2 + b_n^2)}$. Provide the figure and the Matlab code used to obtain it (you can use the `stem` function from Matlab to plot the frequency components). [5 marks]



Calculation and accumulation in the frequency domain:

```
for n = 1:harmonics
    if mod(n,2)
        fn = V0*sin(n*2*pi*f0*t)*(1/n); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V0/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end
```

Plotting in the frequency domain:

```
subplot(2,1,2);
stem(f,An,"filled");%Draw frequency domain
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

For convenience, the frequency and time domains have been calculated simultaneously.

Part B (20 Marks)

B.1) Write *f(t)* in Fourier synthesis form, i.e., as in Equation 2. [4 marks]

We know $a_0=0$, $a_n=0$ and $b_n=2V/\pi n$

$$\boldsymbol{f(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos n2\pi f_0 t + b_n \sin n2\pi f_0 t)}$$

Therefore,

$$\boldsymbol{f(t) = \sum_{n=1}^{\infty}\frac{2V}{\pi n}\sin n2\pi f_0 t}$$

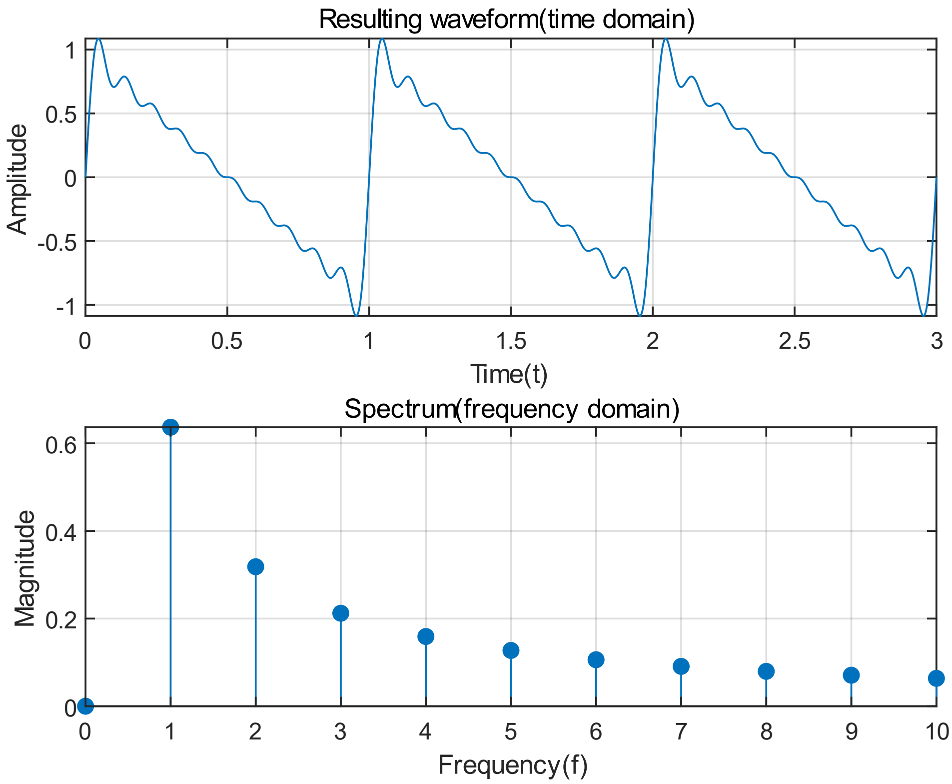
B.2) Calculate the first 10 sinewave coefficients (i.e. b_1, b_2, … , b_{10}). [4 marks]

We know $\boldsymbol{b_n} = \frac{\boldsymbol{2V}}{\boldsymbol{\pi n}}$ and assume V = 1

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
0.6366	0.3183	0.2122	0.1592	0.1273	0.1061	0.0909	0.0796	0.0707	0.0637

B.3) Synthesise the first 10 harmonics of this waveform and plot the result (provide your Matlab code as well). [4 marks]

Waveform of the first 10 harmonics:



Matlab code:

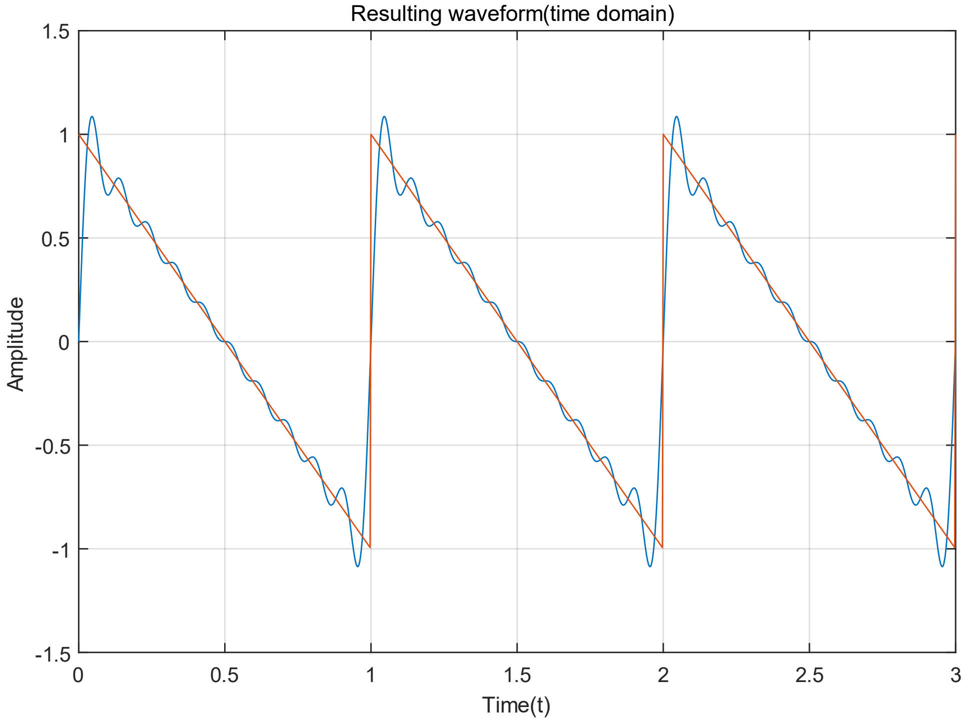
```
clear;
f0 = 1; %fundamental frequency
T = 1/f0;
V = 1;
t = linspace(0,3*T,1000);%Define time domain range and step size
harmonics = 10;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(0,harmonics); %Value range of frequency domain
An(1) = 0;
f = zeros(0,harmonics); %Definition domain of frequency
f(1) = 0;
for n = 1:harmonics
    fn = ((2*V)/(n*pi))*sin(n*2*pi*f0*t); %Harmonic expression
    ft = ft + fn ; %Add to get the total expression
    An(n+1) = (V*2)/(n*pi); %Value in frequency domain
    f(n+1) = n; %Frequency domain
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
% hold on;
% fo=sawtooth(2*pi*t,0); %Original wave
% plot(t,fo);
% hold off;
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

B.4) Plot in the same figure the original and synthesised sawtooth waveforms (provide your Matlab code). Compare the resulting waveform with what you expected to see and discuss the results. [4 marks]

Waveform:



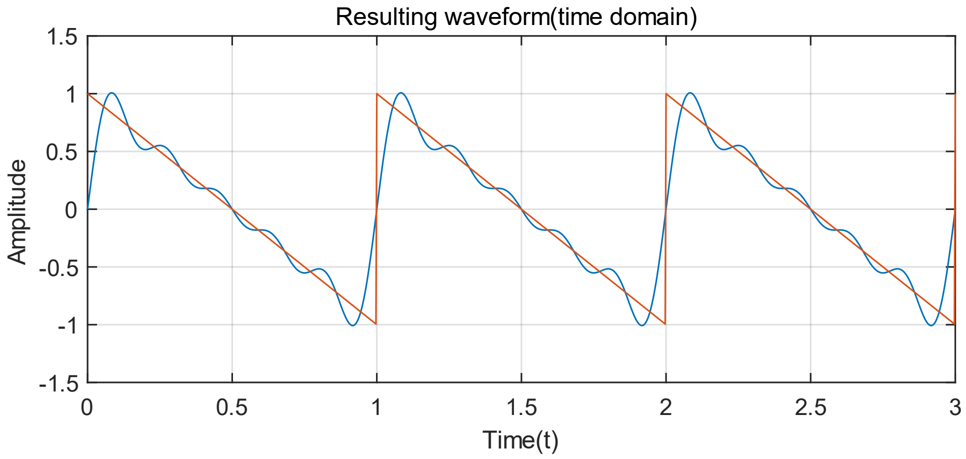
Matlab code:

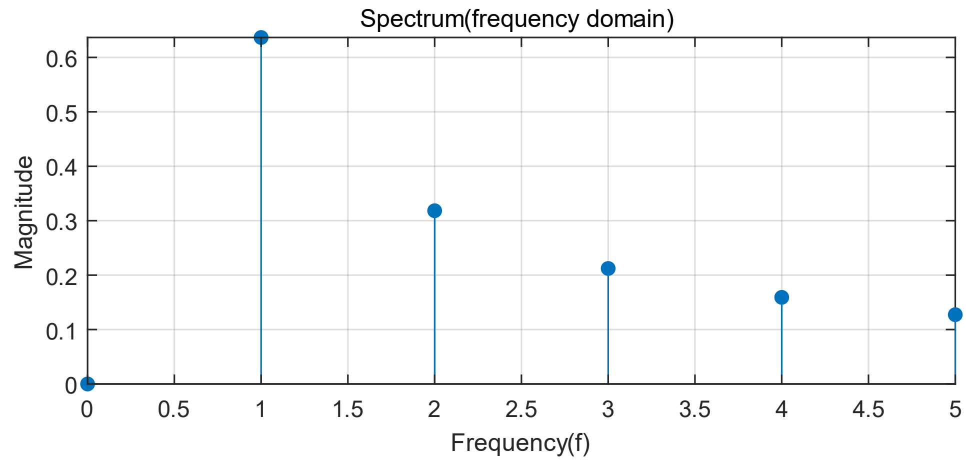
Only a few tweaks to the code were needed to get this result.

```
%subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
hold on;
fo=sawtooth(2*pi*t,0); %Original wave
plot(t,fo);
hold off;
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');
```

As can be seen from the graph, the shape of the original waveform is identical to that provided in the laboratory script, so the original waveform obtained from the simulation is as expected. Furthermore, the shape of both waveforms is close to that of a triangular wave. From the comparison it can be seen that the original waveform shows no ripples and all the lines are straight. The 10th harmonic, on the other hand, shows ripples.

B.5) If the number of harmonics is reduced to 5, comment on the changes that will be observed practically. [4 marks]



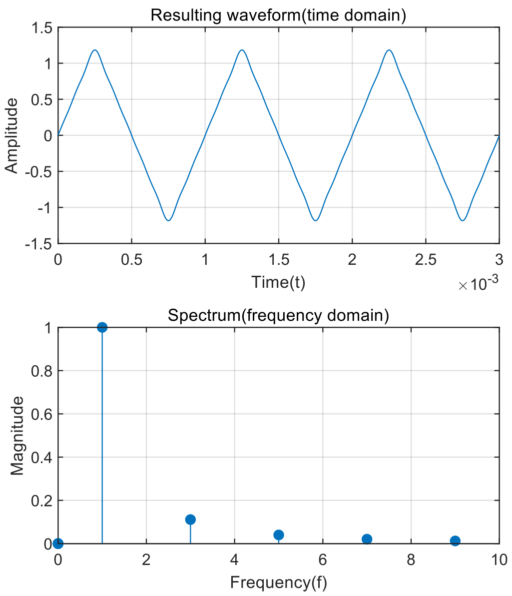


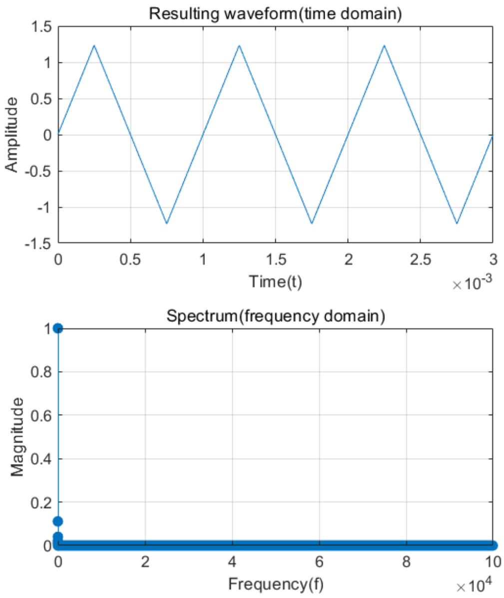
A comparison shows that the ripple phenomenon is greater in amplitude but less rippled when harmonics = 5 than when harmonics = 10. In addition, they are very similar in overall shape. This is because increasing the value of the harmonics reduces the amplitude of the ripple phenomenon and the shape will be more similar to that of a triangle wave. Therefore, it can be concluded that 'when the value of the harmonics tends to infinity, the resulting waveform shape will be very close to the original waveform'.

Part C (15 Marks)

C.1) Synthesise the waveforms below. Plot the resulting waveforms and provide the Matlab code used to obtain the plots as well. [1, 1, 1 and 2 marks, respectively]

Waveform (a):





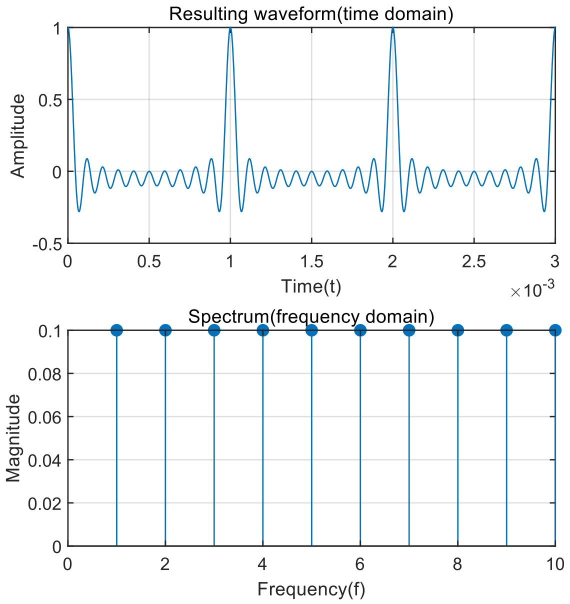
Matlab code (a):

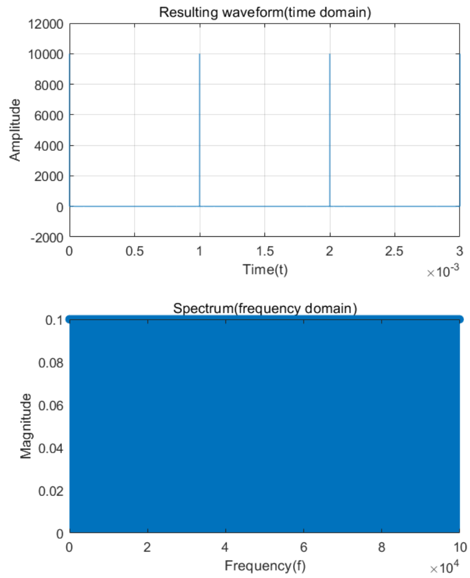
```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 1;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
count = 1;
for n = 1:harmonics
    if mod(n,2)
        count = count+1;
        fn = ((-1)^count)*sin(n*2*pi*f0*t )*(V/n^2); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/(n^2); %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Waveform (b):





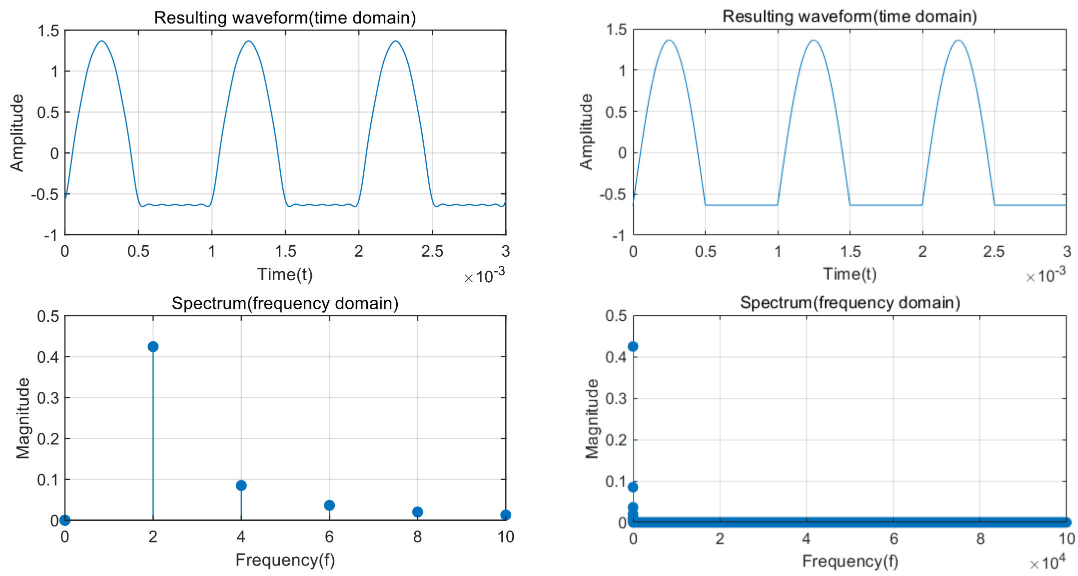
Matlab code (b):

```
clear;
f0 = 1000;  %fundamental frequency
T = 1/f0;
V = 0.1;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 10;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    fn = cos(n*2*pi*f0*t)*V; %Harmonic expression
    ft = ft+fn; %Add to get the total expression
    An(n) = V; %Value in frequency domain
    f(n) = n; %Frequency domain
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Waveform (c):



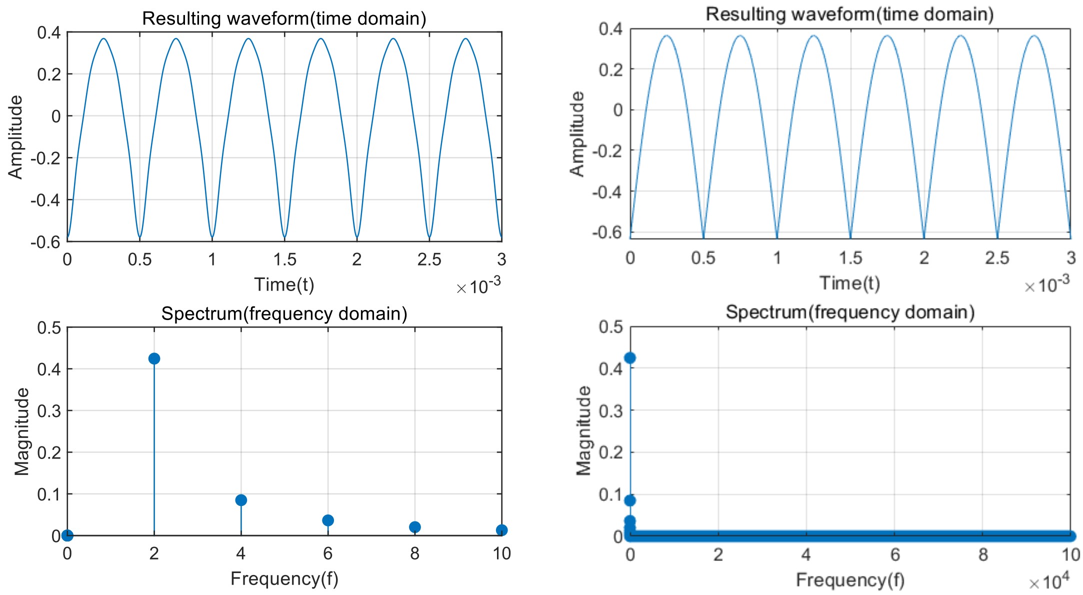
Matlab code (c):

```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 4;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = sin(2*pi*f0*t); %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if ~(mod(n,2))
        fn = -cos(n*2*pi*f0*t )*(V/((n^2-1)*pi)); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/((n^2-1)*pi); %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Waveform (d):



Matlab code (d):

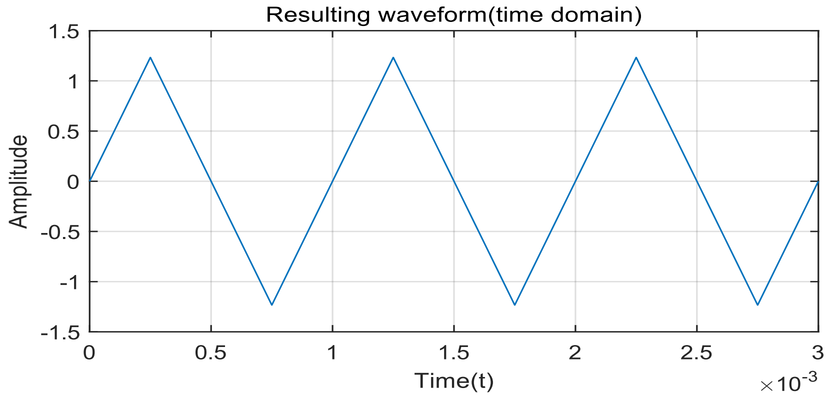
```matlab
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 4;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if ~(mod(n,2))
        fn = -cos(n*2*pi*f0*t )*(V/((n^2-1)*pi)); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/((n^2-1)*pi); %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

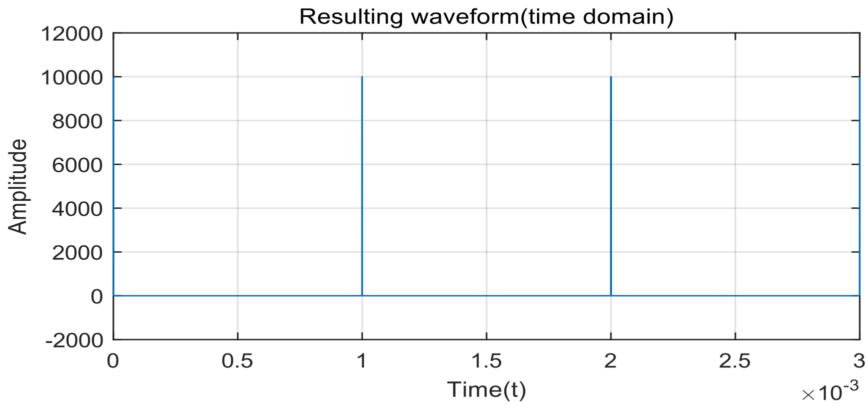
C.2) Comment on your results for each waveform. [3, 3, 2 and 2 marks, respectively]

Result (a):



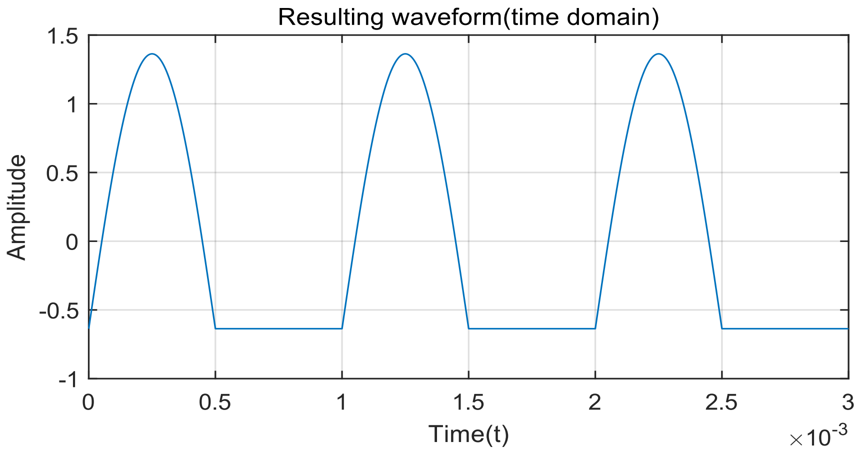
This is a sawtooth waveform, the period of which I set to 1ms. This waveform is observed to be an odd waveform and contains harmonics of odd order. Each waveform has a "sawtooth" shape, with the same amplitude at the beginning and end of the cycle, both being 0. The waveform first rises to a maximum in the first quarter of the cycle, close to 1.2 V. The curve then begins to fall from the first quarter to the third quarter, reaching a minimum of about -1.2 V. The waveform then rises again to 0 V. rises to 0 V. As the harmonics increase, each segment of the resulting waveform becomes progressively straighter and the extent of the ripple phenomenon becomes smaller. As a result, when the harmonics are large enough, the waveform becomes a perfect 'triangle' shape.

Result (b):



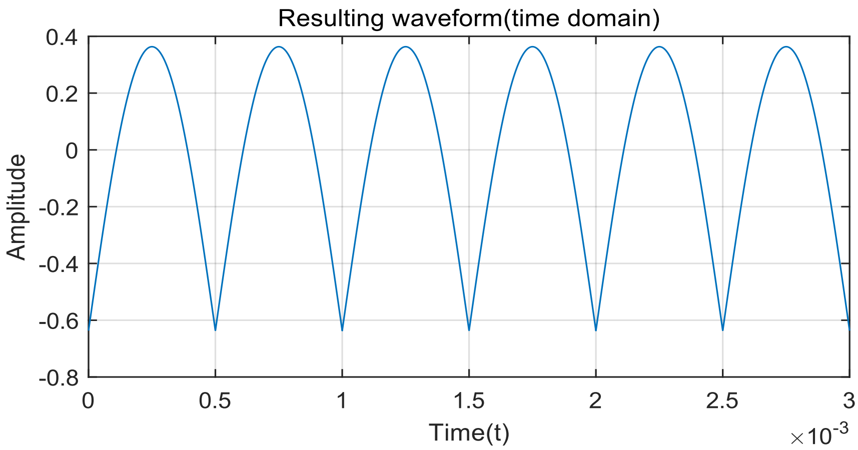
This is an impulsive signal based on the Dirac delta function. The period of this waveform is set to 1 millisecond. By observation it is found to be an even waveform. When the number of harmonics is small, all waveforms fluctuate most of the time during a cycle. The waveform starts to fall from the initial position and for most of the time in between it rises and falls at amplitude = 0V. As the harmonics increase, the amplitude of the ripple in the middle part becomes smaller. When the harmonics are large enough, the overshoot of the waveform will be very small. This is also the Gibbs effect, as the harmonics increase the ripple of the waveform will gradually decrease and will eventually reach a steady state where overshoot and ripple will be difficult to observe.

Result (c):



The period of this waveform is set to 1 millisecond. As can be seen from the graph, the wave appears to be 'clipped' from halfway through the period to the last time in a cycle. Furthermore, the shape of the waveform appears to be in the shape of an 'arc' in the first half of the cycle, but the Gibbs phenomenon appears to fluctuate and overshoot in the second half of the cycle. The maximum amplitude in a cycle is approximately 1.37V and the minimum amplitude is approximately -0.63V.

Result (d):

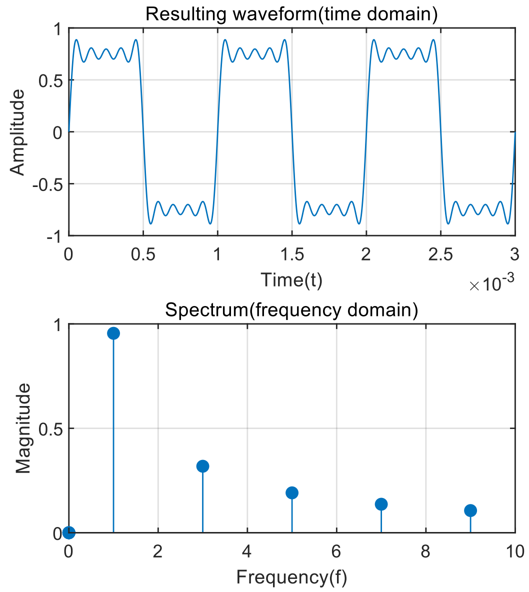


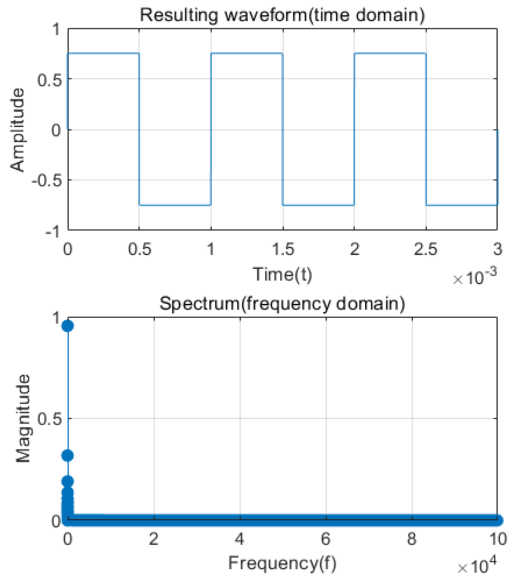
The period of this waveform was set to 1 millisecond. The waveform has a maximum amplitude of approximately 0.37 V and a minimum amplitude of approximately -0.58 V over one cycle. there is no Gibbs phenomenon in the second half of the waveform where there are no fluctuations compared to the waveform without the sine off. In cycles 0-1/4 the waveform rises and in cycles 1/4-1/2 the waveform falls from its highest point to its lowest point. Furthermore, at cycles 1/2-3/4, the waveform rises again to its highest point, while at cycles 3/4-1, the waveform falls again from its highest point to its lowest point.

Part D (15 Marks)

D.1) Synthesise and plot this square wave (provide your Matlab code as well). Calculate the percentage overshoot of the synthesised waveform (compared with the ideal waveform) at the discontinuity. How does this compare with the expected limit of 17.9%? [2 marks]

Waveform:





Matlab code:

```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 3/pi;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if mod(n,2)
        fn = sin(n*2*pi*f0*t)*(V/n); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

we know:

$$Percentage = \frac{V_{max} - V_{steady}}{V_{steady}}$$

So, we did this by performing two simulations, one containing the 10th harmonic and the other containing the 100,000th harmonic. By the peak of the 10th harmonic, we find $V_{max} \approx 0.8867$ and by the peak of the 100,000th harmonic (which is almost stable) we find $V_{steady} \approx 0.75$, the final calculation gives an overshoot percentage of approximately 18.23%. In addition, the simulated 100th harmonic gives Vmax = 0.8842, which gives a percentage overshoot of approximately 17.89%. As the expected limit is 17.9%, the percentage obtained will be greater than 17.9% when harmonics = 10, indicating a relatively large overshoot value, which means that waveform fluctuations will be more pronounced when the harmonics are relatively small. From the above results it can be seen that the expected value of 17.9% is most closely approached when the number of harmonics is around 950. As the harmonics increase, the percentage overshoot will gradually converge to 0%, which is as expected. This is because as the harmonics gradually increase, the ripple will become smaller, which leads to smaller overshoot values and a stabilisation of the waveform.

D.2) What is the name of this overshoot? Explain it. [2 marks]

Gibbs phenomenon, the Gibbs phenomenon reflects the difficulty of approximating discontinuous functions with a finite series of continuous sine and cosine waves. Although each partial sum of a Fourier series exceeds the function it approximates, the limit of the partial sum does not. Furthermore, as the number of summation terms increases, the closer the value of x that reaches maximum overshoot approaches the discontinuity. From a signal processing perspective, the Gibbs phenomenon is the step response of a low-pass filter, but the overall integral does not change under filtering. Sinusoidal integration leads to overshoot, which is related to the Gibbs phenomenon. The sinusoidal integral can be thought of as the convolution of the sinc function with the step function, which is related to the truncated Fourier series.

D.3) Give the Fourier series in each case for the resulting waveform if the above square wave is used as an input for:

D.3) (a) A low-pass filter with gain and phase responses as given in Figures 8 and 10 respectively. [1 mark]

For harmonics of infinite number, its Fourier series is:

$$\boldsymbol{f(t) = \sum_{n=1}^{\infty} \frac{3}{n\pi} \sin(n2\pi f_0 t - \frac{n\pi}{2})}$$

After simplification we get:

$$f(t) = \sum_{n=1}^{\infty} (-1^{\frac{n+1}{2}}) \frac{3}{n\pi} \cos(n2\pi f_0 t)$$

It is easy to observe that this is an even wave and contains only odd harmonics.

D.3) (b) A low-pass filter with gain and phase responses as given in Figures 8 and 11 respectively. [1 mark]

For harmonics of infinite number, its Fourier series is:

$$f(t) = \sum_{n=1}^{\infty} \frac{3}{n\pi} \sin(n2\pi f_0 t - \frac{\pi}{2})$$

After simplification we get:

$$f(t) = \sum_{n=1}^{\infty} \frac{-3}{n\pi} \cos(n2\pi f_0 t)$$

It is easy to observe that this is an even wave and contains only odd harmonics.

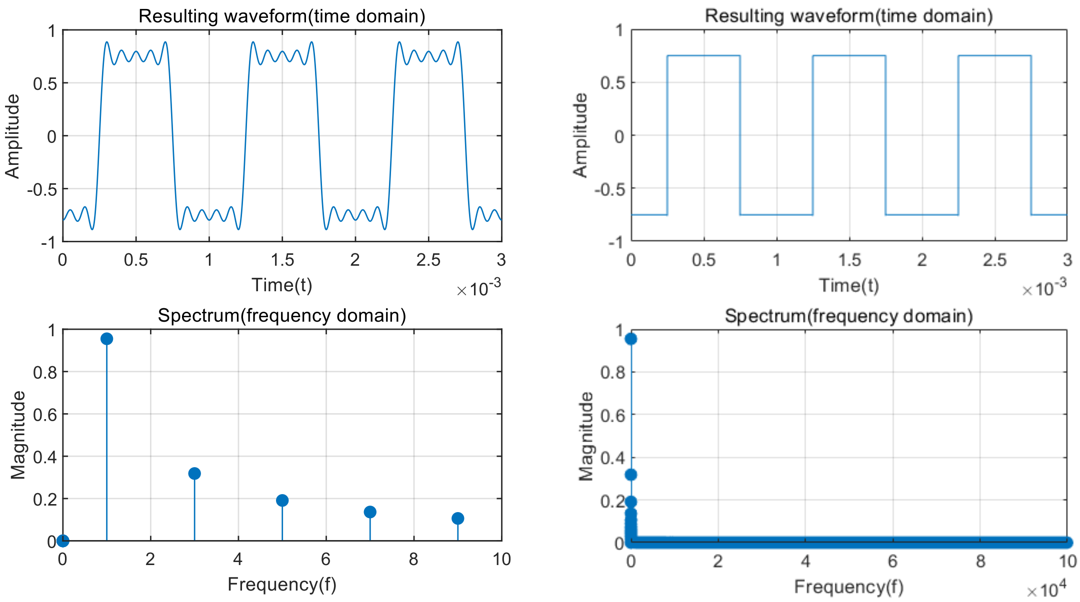
D.3) (c) A band-pass filter with gain and phase responses as given in Figures 9 and 10 respectively. [1 mark]

For harmonics of finite number, his Fourier series is:

$$f(t) = \frac{3}{\pi} (\frac{1}{6} \sin\left(6\pi f_0 t - \frac{3\pi}{2}\right) + \frac{1}{5} \sin\left(10\pi f_0 t - \frac{5\pi}{2}\right) + \frac{1}{7} \sin(14\pi f_0 t - \frac{7\pi}{2}) + \frac{1}{18} \sin(18\pi f_0 t - \frac{9\pi}{2}))$$

D.4) Synthesise the above waveforms and draw the obtained waveforms for filters (a), (b) and (c), providing the Matlab code as well. [3 marks]

Waveform (a):



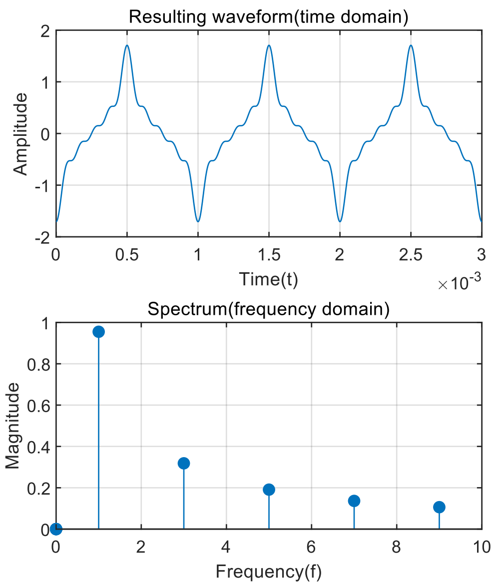
Matlab code (a):

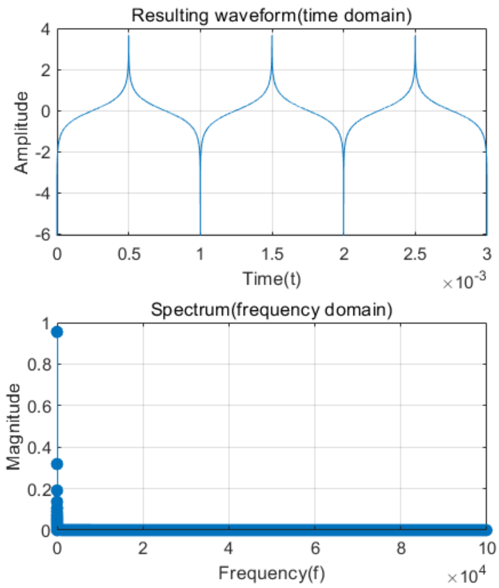
```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 3/pi;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if mod(n,2)
        fn = sin(n*2*pi*f0*t-(n*pi)/2)*(V/n); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Waveform (b):





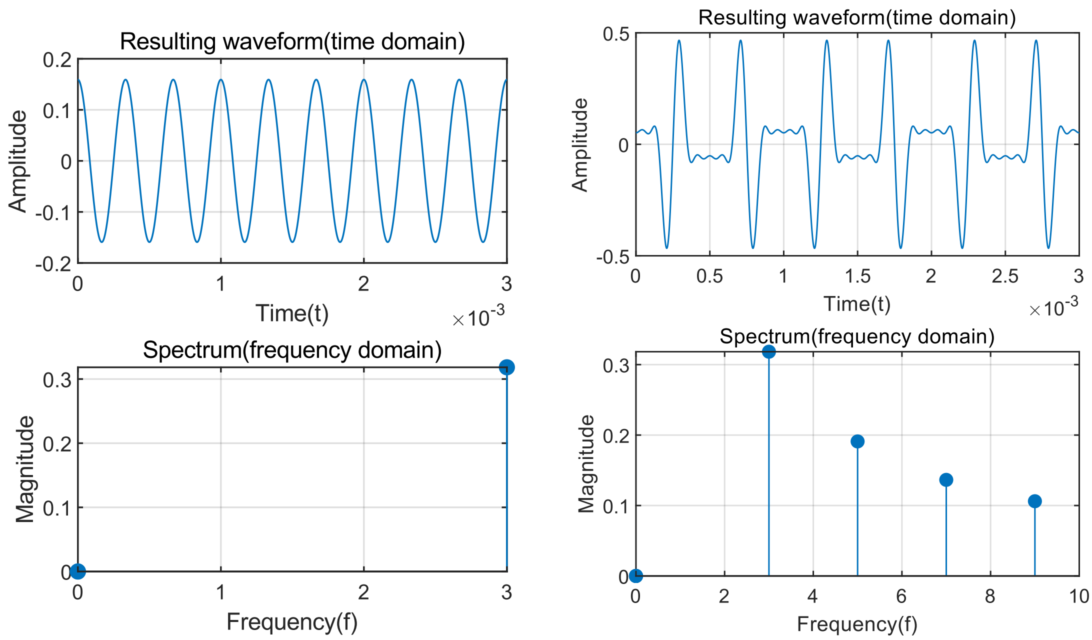
Matlab code (b):

```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 3/pi;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 100000;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 1:harmonics
    if mod(n,2)
        fn = sin(n*2*pi*f0*t-pi/2)*(V/n); %Harmonic expression
        ft = ft+fn; %Add to get the total expression
        An(n) = V/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

Waveform (c):



Matlab code (c):

```
clear;
f0 = 1000; %fundamental frequency
T = 1/f0;
V = 3/pi;
t = linspace(0,3*T,10000);%Define time domain range and step size
harmonics = 10;%Number of harmonics
ft = 0; %Define initial waveform
An = zeros(1,harmonics); %Value range of frequency domain
f = zeros(1,harmonics); %Definition domain of frequency
for n = 3:harmonics
    if mod(n,2)
        if ~mod(n,3)
            fn = sin(n*2*pi*f0*t-(n*pi)/2)*(V/(2*n)); %Harmonic expression
        else
            fn = sin(n*2*pi*f0*t-(n*pi)/2)*(V/n); %Harmonic expression
        end
        ft = ft+fn; %Add to get the total expression
        An(n) = V/n; %Value in frequency domain
        f(n) = n; %Frequency domain
    end
end

subplot(2,1,1);%Draw time domain waveform
plot(t,ft);
grid on;
xlabel('Time(t)');
ylabel('Amplitude');
title('Resulting waveform(time domain)');

subplot(2,1,2); %Draw frequency domain
stem(f,An,"filled");
grid on;
xlabel('Frequency(f)');
ylabel('Magnitude');
title('Spectrum(frequency domain)');
```

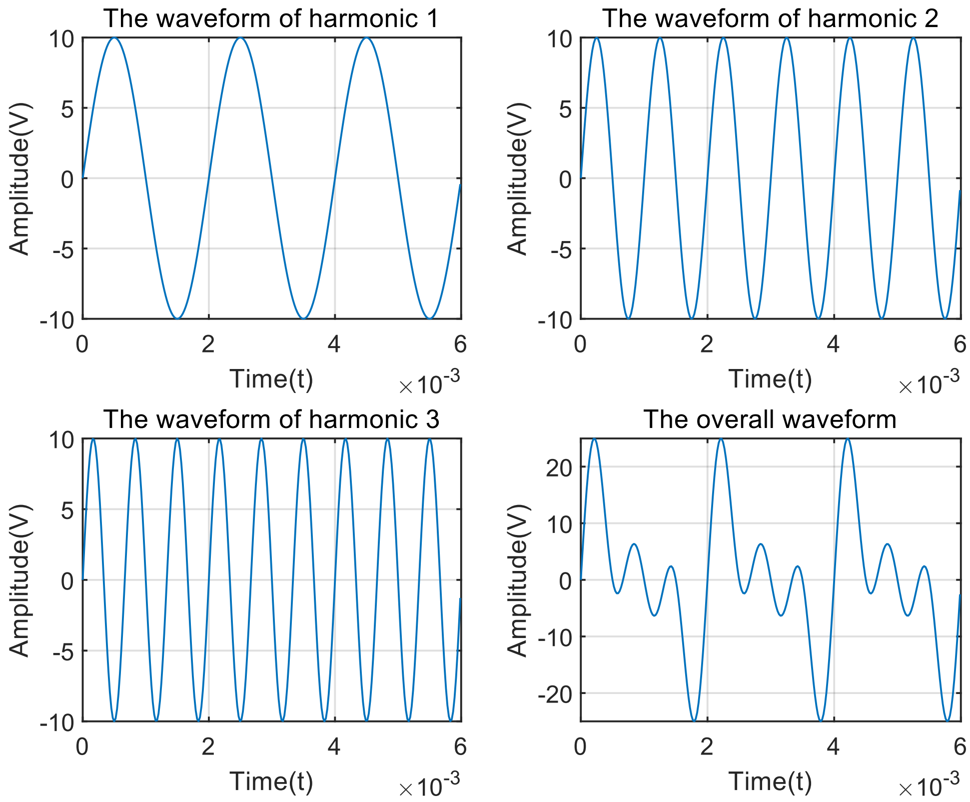
D.5) What is the fundamental frequency of the output from filter (c)? Why? [5 marks]

The fundamental frequency is still 1kHz. Adding a bandpass filter and increasing the phase shift will not affect the fundamental frequency. Since the fundamental frequency in the experiment is 1kHz, the final output signal will have a fundamental frequency of 1 KHz. from another point of view, before defining the fundamental frequency, it is necessary to define the fundamental period, which is the minimum period on the function. The fundamental frequency is defined as its reciprocal. The minimum period of the function does not change and therefore the fundamental frequency does not change. Furthermore, the bandpass filter is only used to filter the frequency and does not affect the fundamental frequency. For the phase filter response, it only changes the phase of the function, independent of the frequency, so adding a phase filter response does not change the value of the fundamental frequency.

Part E (10 Marks)

E.1) For the wave in equation 15, listen to harmonics 1, 2 and 3 individually then as a chord. Plot the chord waveform as well. Provide the Matlab code used to listen to the harmonics/chord and plot the chord. [3 marks]

Waveform of harmonics:



Matlab code:

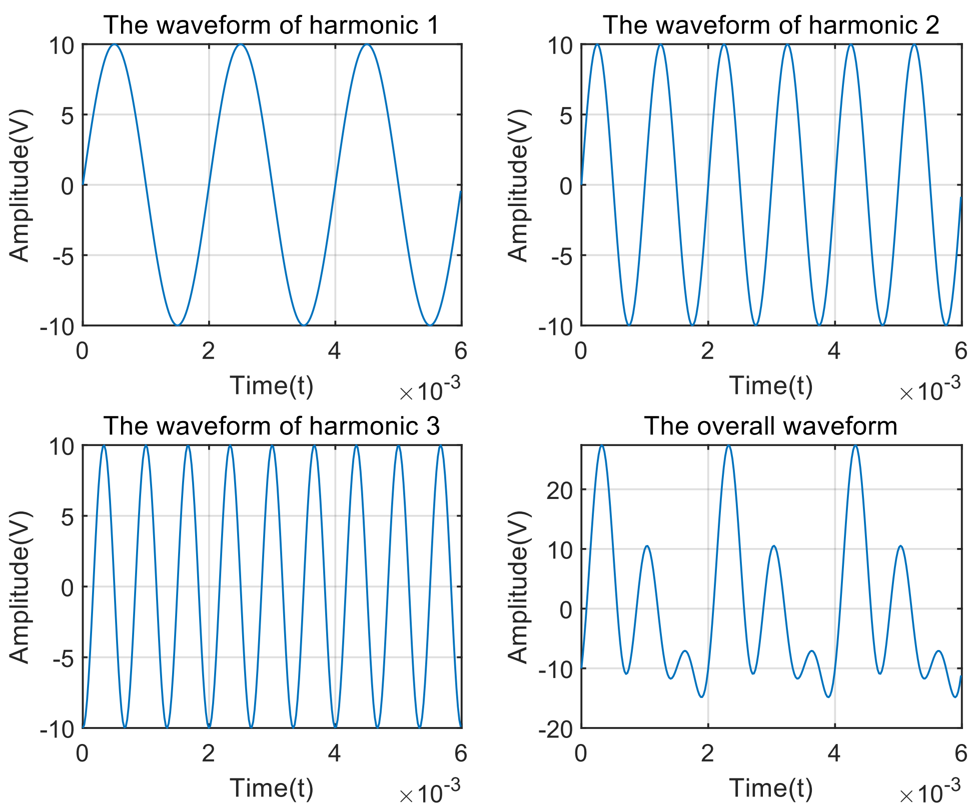
```
%Harmonic 1
fs = 44100;
duration = 3;
t = 0:1/fs:duration;
amplitude = 10;
f = 500;
Tmax = 3;
wave1 = amplitude*sin(2*pi*f*t);
sound(wave1,fs);
pause(duration);
clippedt = t(1:find(t <= Tmax/f, 1,'last'));
clippedwave = wave1(1:find(t <= Tmax/f, 1,'last'));
subplot(2,2,1);
plot(clippedt, clippedwave);
grid on;
xlabel('Time(t)');
ylabel('Amplitude(V)');
title('The waveform of harmonic 1');
```

```
%Harmonic 2
wave2 = amplitude*sin(2*pi*2*f*t);
sound(wave2,fs);
pause(duration);
clippedt = t(1:find(t <= Tmax/f,1 ,'last'));
clippedwave = wave2(1:find(t <= Tmax/f, 1,'last'));
subplot(2,2,2);
plot(clippedt, clippedwave);
grid on;
xlabel('Time(t)');
ylabel('Amplitude(V)');
title('The waveform of harmonic 2');

%Harmonic 3
wave3 = amplitude*(sin(2*pi*3*f*t));
sound(wave3,fs);
pause(duration);
clippedt = t(1:find(t <= Tmax/f, 1,'last'));
clippedwave = wave3(1:find(t <= Tmax/f, 1,'last'));
subplot(2,2,3);
plot(clippedt, clippedwave);
grid on;
xlabel('Time(t)');
ylabel('Amplitude(V)');
title('The waveform of harmonic 3');
%Sum of harmonics
wave4 = amplitude*(sin(2*pi*f*t)+sin(2*pi*2*f*t)+sin(2*pi*3*f*t));
sound(wave4,fs);
pause(duration);
clippedt = t(1:find(t <= Tmax/f, 1,'last'));
clippedwave = wave4(1:find(t <= Tmax/f, 1,'last'));
subplot(2,2,4);
plot(clippedt, clippedwave);
grid on;
xlabel('Time(t)');
ylabel('Amplitude(V)');
title('The overall waveform');
```

E.2) Alter the phase of the third harmonic in equation 15 by 90° and repeat the tasks in the point above. [3 marks]

Waveform of harmonics:



Matlab code:

As only the phase of the third harmonic has been changed and the rest of the code is exactly the same as before, only the changed parts are shown here.

```
%Harmonic 3
wave3 = amplitude*(sin(2*pi*3*f*t-pi/2));
%wave3 = amplitude*(sin(2*pi*3*f*t));
sound(wave3,fs);
```

```
%Sum of harmonics
wave4 = amplitude*(sin(2*pi*f*t)+sin(2*pi*2*f*t)+sin(2*pi*3*f*t-pi/2));
%wave4 = amplitude*(sin(2*pi*f*t)+sin(2*pi*2*f*t)+sin(2*pi*3*f*t));
sound(wave4,fs);
```

E.3) Discuss the effect of altering the phase of the third harmonic, both on the sound and plot of the chord. Do the sound or plot change as you alter the phase? Why? [2 marks]

1. The effect of sound

For the waveform with the added phase shift, the difference is not significant, although there is a small difference compared to the two sounds. In this case, there are no non-linear effects. In addition, because the signal period is repetitive and of relatively short duration. Adding the phase shift may vary slightly. However, depending on the frequency of repetition and the nature of the pulse, this variation may be extremely weak to the listener. For chords, it is also difficult to hear the difference with the human ear. Although the total waveform looks different, the volume is reduced so that the non-linear effects are not apparent and one hears very similar sounds.

2. The effect of plot

By obtaining two sets of graphs, it can be seen that the two chords have different shapes, but the differences are not very pronounced. The amplitude of each ripple is not the same. In addition, the maximum and minimum values of the waveforms are not the same. When the phase shift is not set, the maximum and minimum values of the curve are 24.95 V and -24.99 V. When the phase shift is set, the maximum and minimum values of the curve are: 27.4 V and -14.80 V respectively.

E.4) What does the above tell you about the human ear? [2 marks]

The human ear can hear sounds from 20-20,000 Hz, but the phase shift of the sound is not clearly audible. For some people who are extremely sensitive to sound the difference may be audible. In addition, when a phase shift occurs, the shape of the chord waveform will change. This phenomenon can be seen in the resulting waveform above. This is because the non-linear effect is not obvious. The final chord is produced by the sum of three different sets of sounds. Mixing the three sounds will make the sound more difficult to hear the difference. This experiment tells me that it is difficult for the human ear to clearly distinguish the differences in sound caused by phase shifts.