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]

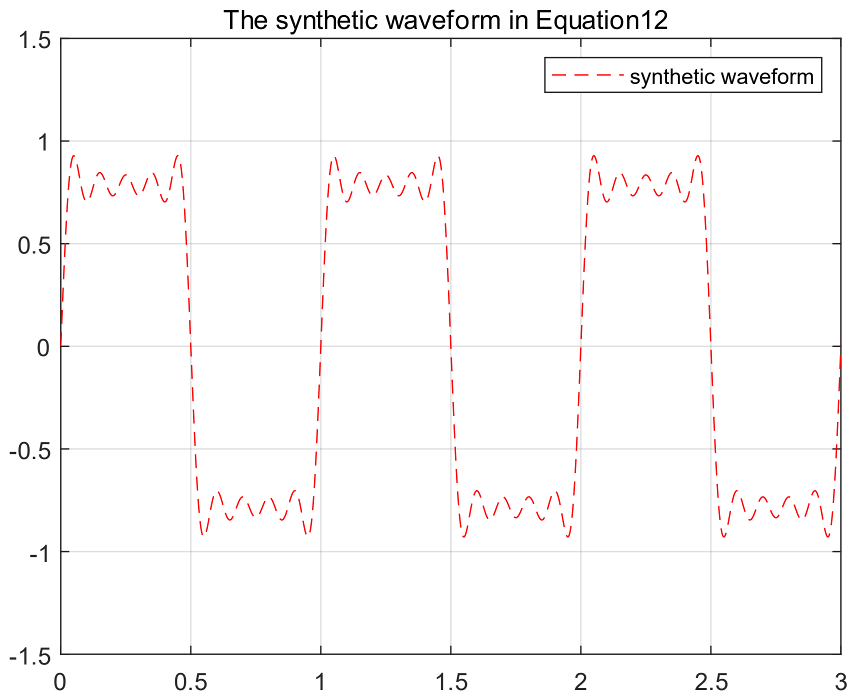

Figure 1. The resulting waveform

$$f(t) = V_0\left(sin\omega t + \frac{1}{3}sin3\omega t + \frac{1}{5}sin5\omega t + \frac{1}{7}sin7\omega t + \frac{1}{9}sin9\omega t\right) \qquad (12)$$

$$\mathrm{T} = 1\mathrm{s}, \omega = \frac{2\pi}{\mathrm{T}} = 2\pi, V_0 = 1$$

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

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

Peak-to-peak amplitude:
According to Fig.1, peak-to-peak amplitude is 1.8752V.

Symmetry:
According to equation 12, the resulting waveform is an even function, which symmetry about the origin and repeat every 1 s.

Ripple:
As it is a periodic waveform, the ripple can be discussed in only 1 period (0<t<1). As shown in Fig.1, the ripple voltage is about 0.23V, and there is also a degree of overshoot. And at the discontinuous point such as t = 0.5, 1, 1.5, 2…, there is an obvious Gibbs phenomenon.

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 infinite harmonics were available, the shape of the resulting waveform would be a perfect square wave. It can be seen in Fig.2 that when n=9999 (can be regarded as an unlimited number), the synthetic waveform is very close to a perfect square wave.

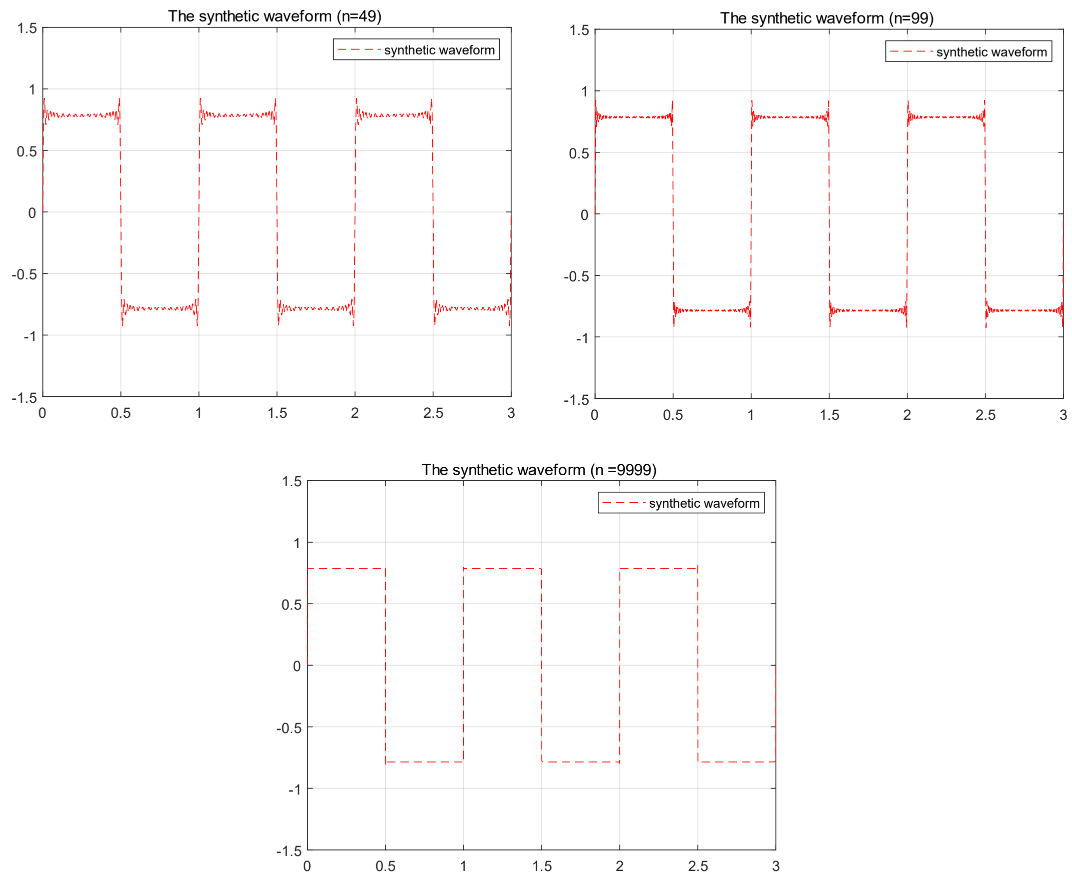

Figure 2. The synthetic waveform with different n (the number of harmonics)

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

$$a_0 = \frac{1}{\mathrm{T}}\int_0^{\mathrm{T}} f(t)\,\mathrm{d}t = \frac{1}{1}\int_0^{1} f(0<t<0.5) + f(0.5<t<1)\,\mathrm{d}t = 0$$

As this synthetic waveform is symmetry about the origin and repeat every 1 s, $f(0 < t < 0.5) = \text{-}f(0.5 < t < 1)$.

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

$$a_n = 0$$
$$b_n = \frac{V_0}{n} \cdot \frac{1 + (-1)^{n+1}}{2}$$

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]

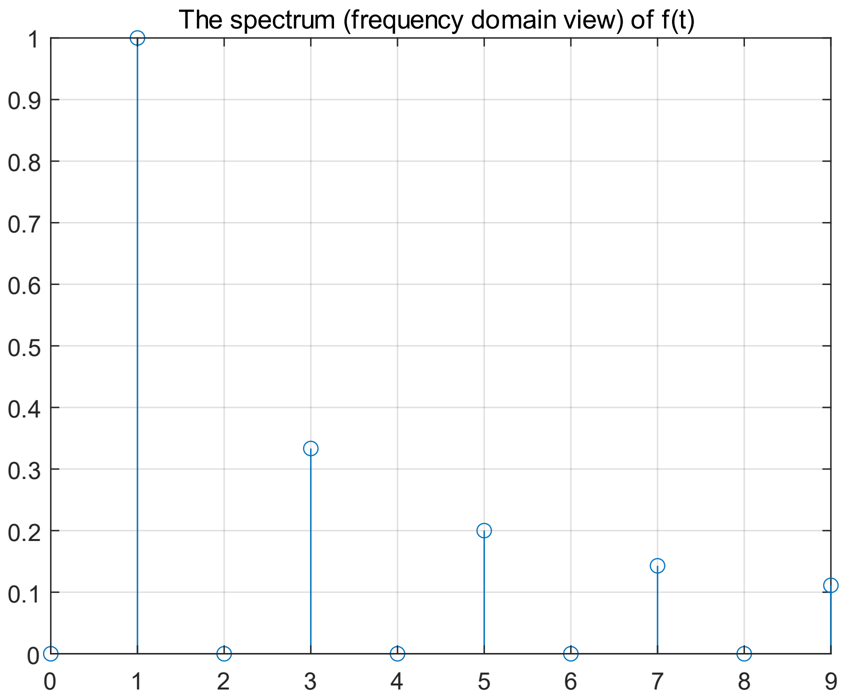

Figure 3. Frequency domain view of f(t)

Part B (20 Marks)

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

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

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

Set V = 1,

Table 1. The first 10 sinewave coefficients

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
$\frac{2}{\pi}$	$\frac{1}{\pi}$	$\frac{2}{3\pi}$	$\frac{1}{2\pi}$	$\frac{2}{5\pi}$	$\frac{1}{3\pi}$	$\frac{2}{7\pi}$	$\frac{1}{4\pi}$	$\frac{2}{9\pi}$	$\frac{1}{5\pi}$

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

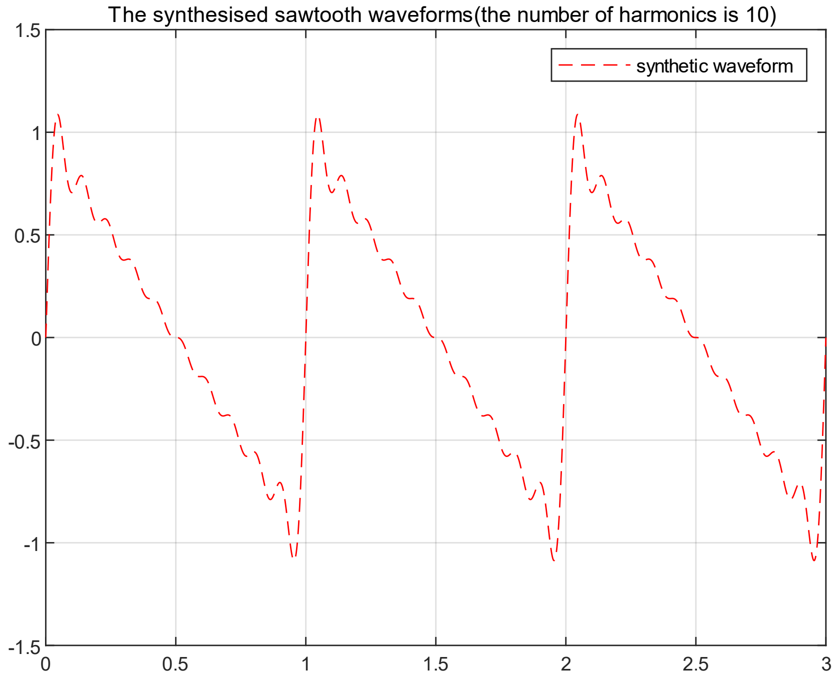

Figure 4. The synthesised waveform

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

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]

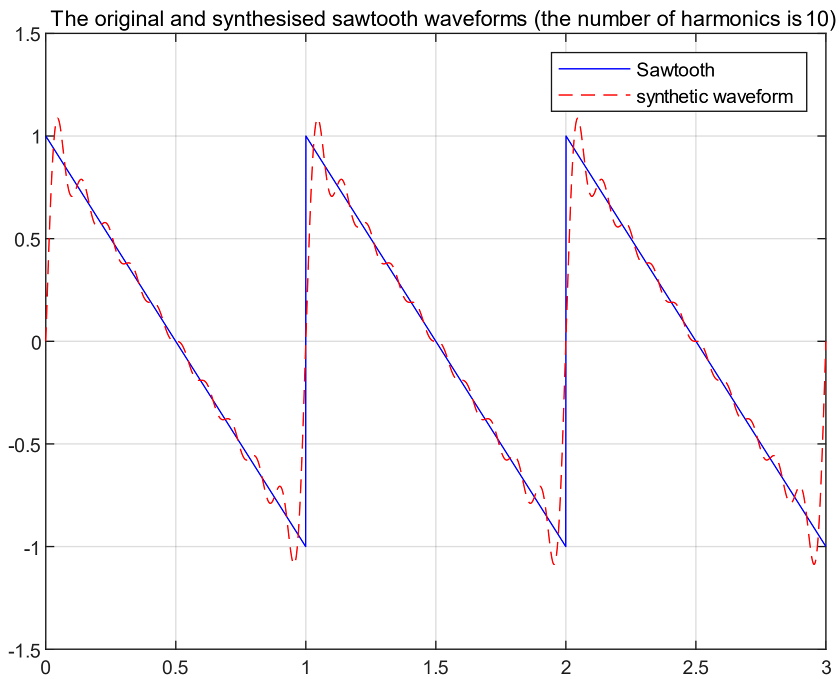

Figure 5. The synthesised and original waveform (n=10)

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program. Compared to the original signal, the synthesised signal has a larger peak-to-peak value (there is a degree of overshoot) and the presence of ripples. In general, the shape and trend of the synthesised signal is essentially the same as the original signal.

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

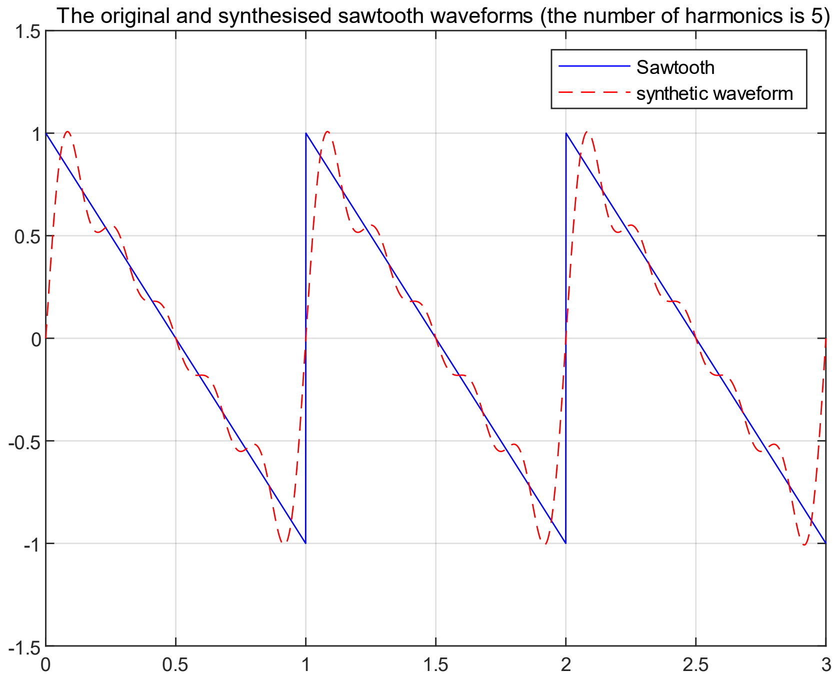

Figure 6. The synthesised and original waveform (n=5)

As n is reduced to 5, the Gibbs phenomenon in the synthesised signal are more obvious, but the peak-to-peak value is reduced, that is, the degree of overshoot decreases. However, in general, the synthetic signal at n=5 is more divergent from the original signal compared to the synthetic signal at n=10, but still follows the trend of the original signal shape.

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]

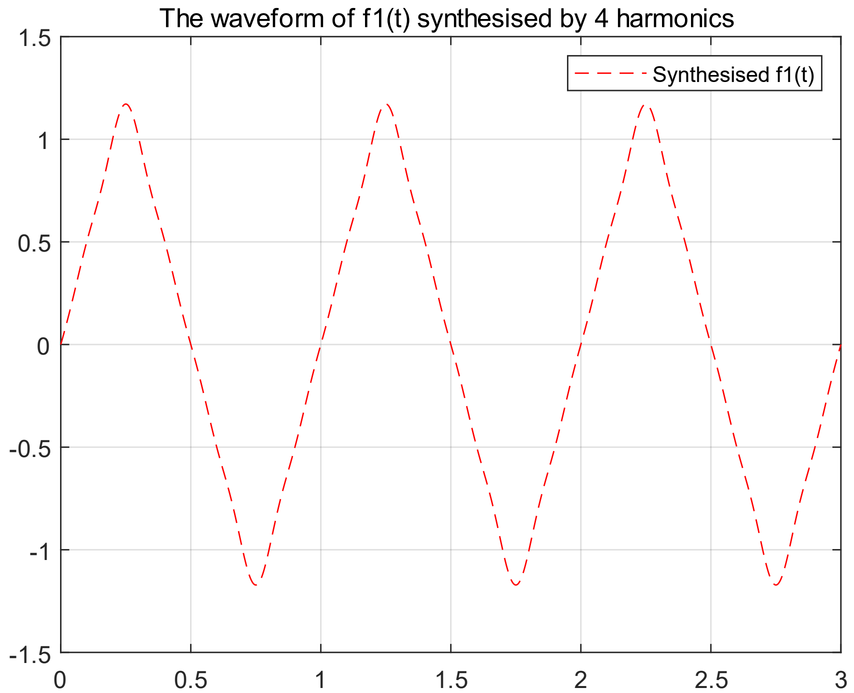

Figure 7. The resulting waveform of (a)

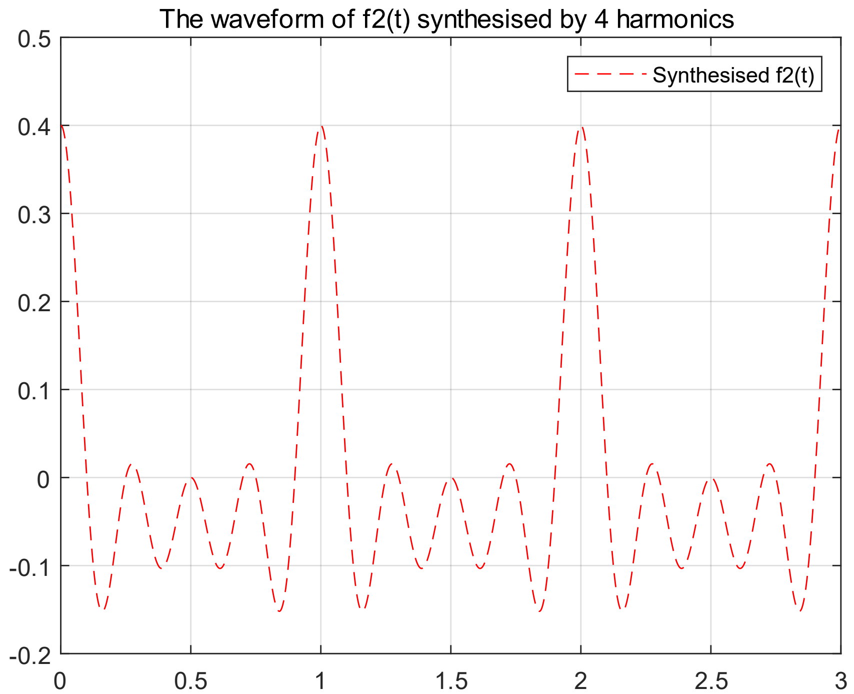

Figure 8. The resulting waveform of (b)

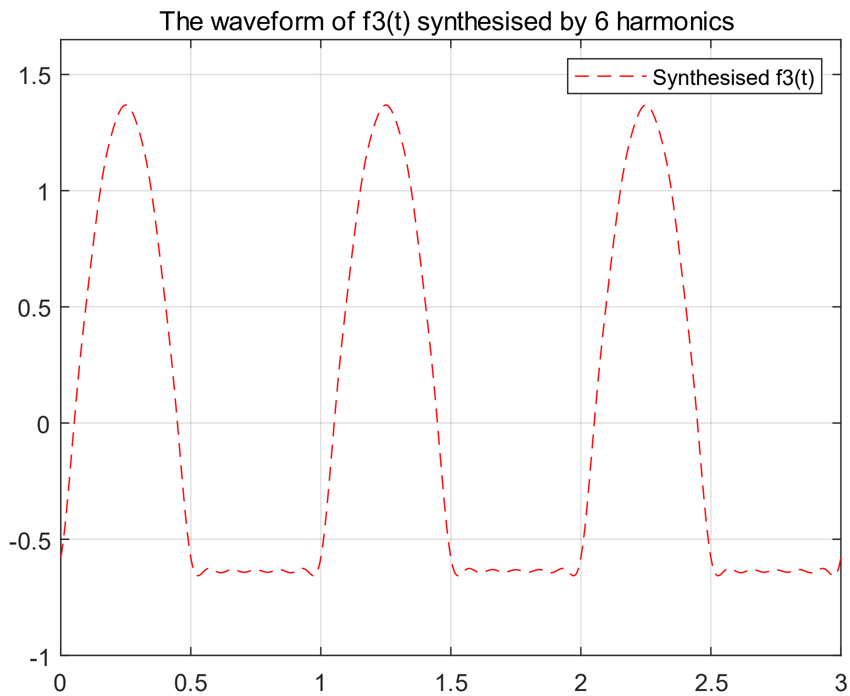

Figure 9. The resulting waveform of (c)

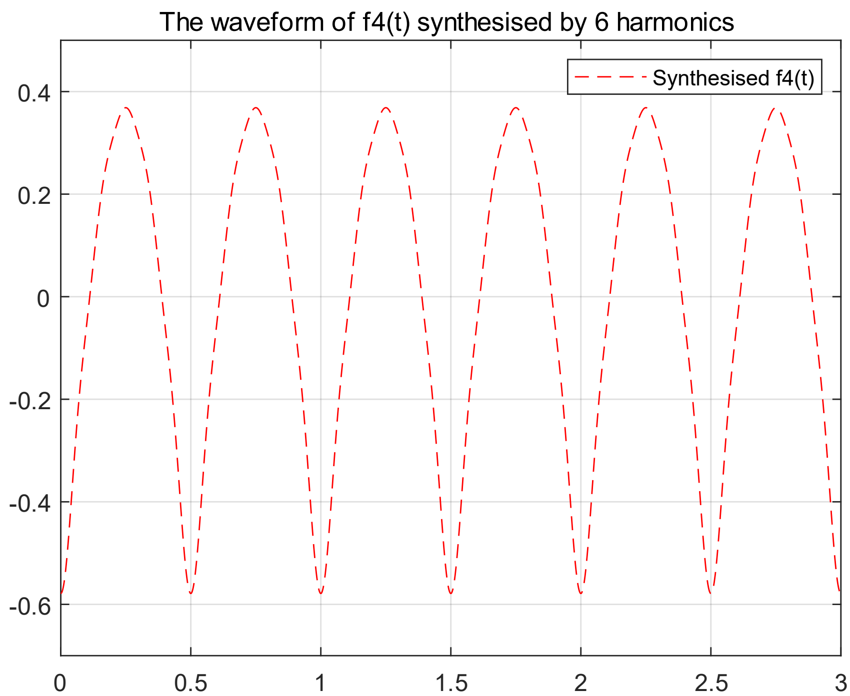

Figure 10. The resulting waveform of (d)

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

C.2) Comment on your results for each waveform. [3, 3, 2 and 2 marks, respectively]
a) This synthesis is periodic, T = 1. As shown in Figure 7, and when there are only four harmonics, the resulting waveform is already very close to the target triangle waveform.

However, the peak-to-peak value of the resulting waveform is slightly smaller than that of the target triangle waveform, and there is no ripple effect and overshoot.

b) The resulting waveform is a periodic signal, T = 1. The maximum value of this synthesis signal equals to $\frac{the\ number\ of\ harmonics}{10}$. As shown in Fig.8, the maximum value of this waveform is 0.4, and the number of harmonics is 4. And when n = 4, the ripple voltage is approximately -0.1V. There also exists overshoot. When n tends to be infinite, the waveform will look like an impulse.

c) The resulting waveform is a periodic signal, T = 1. And when n tends to be infinite, the synthesis waveform satisfies a clipping sine signal. As shown in Fig.9, there are little ripples in the clipping part (e.g., 0.5<t<1), and the ripple voltage is only about 0.01V, as well as a relatively small overshoot.

d) Compared to the resulting waveform in c), the harmonic of $sin(\omega t)$ is excluded in d), so there is no clipping part. However, it is still a periodic waveform and T = 0.5, which is the half of that in waveform c). There are no overshoot phenomenon and ripples.

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]

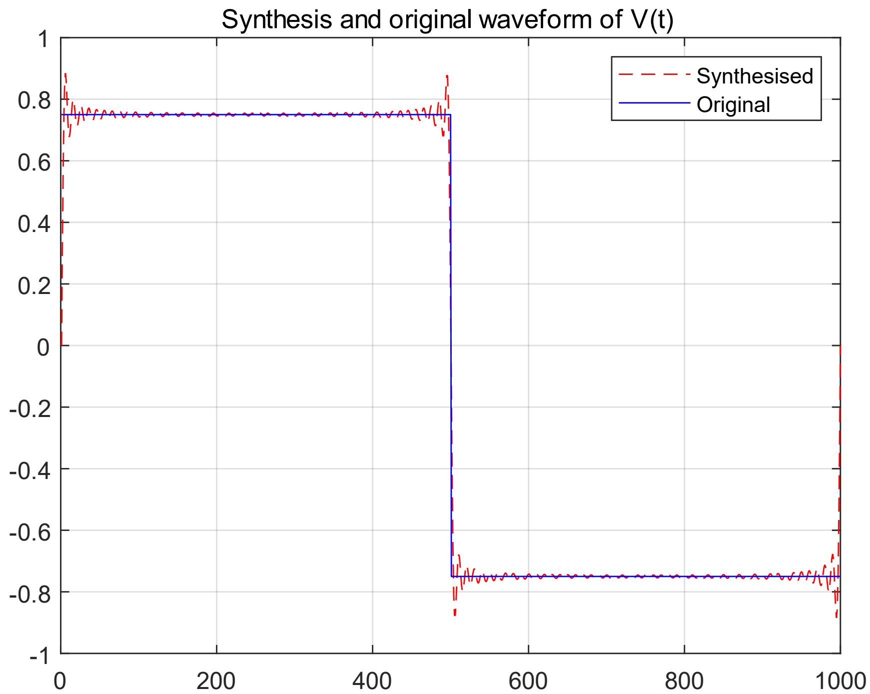

Figure 11. Synthesis (n = 99) and original square wave

The percentage overshoot:

```
ans =

    'Percentage overshoot is 17.901%'

fx >>
```

Figure 12. Screenshot of percentage overshoot

The percentage overshot when n = 99 is 17.901%, which is very close to the expected limit of 17.9%.

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

D.2) What is the name of this overshoot? Explain it. [2 marks]
The name of this overshoot is Gibbs phenomenon. Gibbs phenomenon can be explained as oscillations at waveform discontinuity points, as shown in Fig.11, when t = 0, 500, 1000, the maximum value of synthesis waveform is greater than the ideal one.

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]

$$f(n) = \frac{3}{\pi}[\frac{sin(\omega t - \frac{\pi}{2})}{1} + \frac{sin(3\omega t - \frac{3\pi}{2})}{3} + \frac{sin(5\omega t - \frac{5\pi}{2})}{5} + \frac{sin(7\omega t - \frac{7\pi}{2})}{7} + \frac{sin(9\omega t - \frac{9\pi}{2})}{9}], \omega = 2\pi f_0$$

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

$$f(n) = \frac{3}{\pi}[\frac{sin(\omega t - \frac{\pi}{2})}{1} + \frac{sin(3\omega t - \frac{\pi}{2})}{3} + \frac{sin(5\omega t - \frac{\pi}{2})}{5} + \frac{sin(7\omega t - \frac{\pi}{2})}{7} + \frac{sin(9\omega t - \frac{\pi}{2})}{9}], \omega = 2\pi f_0$$

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

$$f(n) = \frac{3}{\pi}[\,0 \cdot sin(\omega t - \pi) + \frac{1}{3} \cdot sin(3\omega t - \frac{3}{2}\pi) + \frac{1}{5} \cdot sin\,(5\omega t - \frac{5}{2}\pi) + \frac{1}{7} \cdot \sin\,(7\omega t - \frac{7}{2}\pi) + \frac{1}{9} \cdot \sin\,(9\omega t - \frac{9}{2}\pi)$$

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]

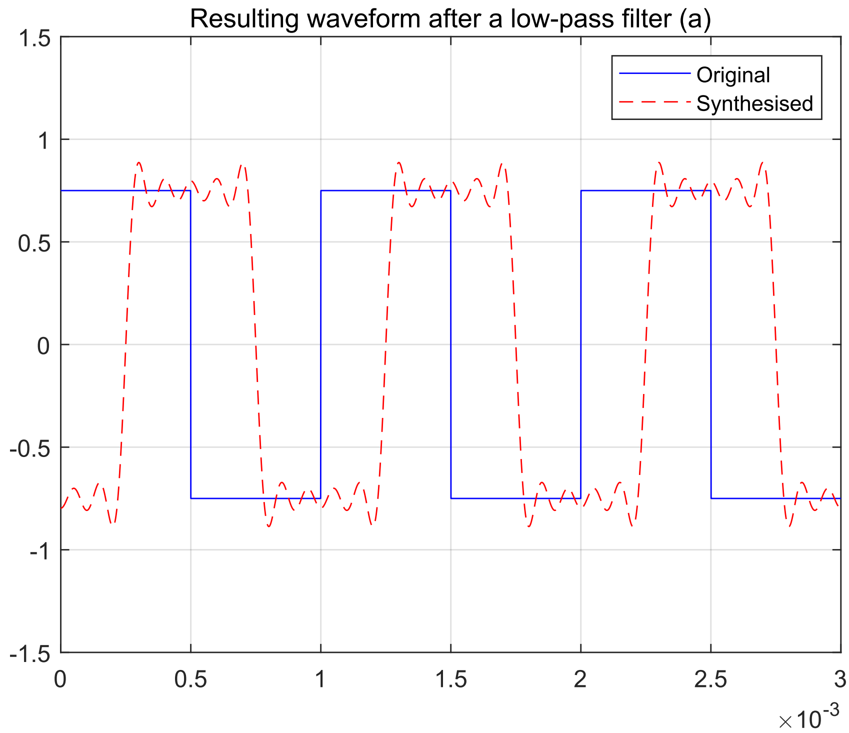

Figure 13. Resulting waveform after filter (a)

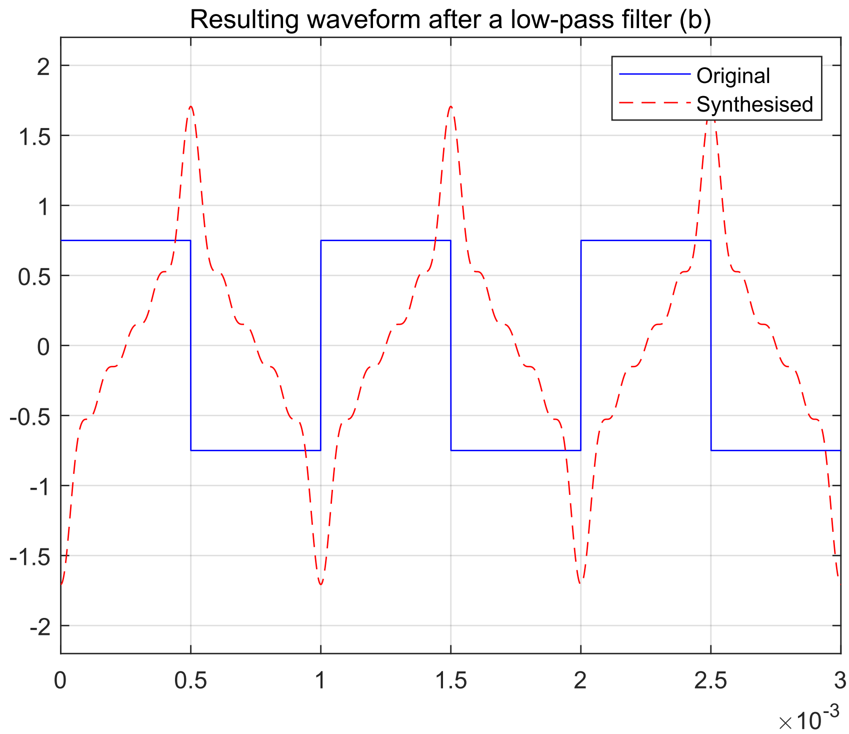

Figure 14. Resulting waveform after filter (b)

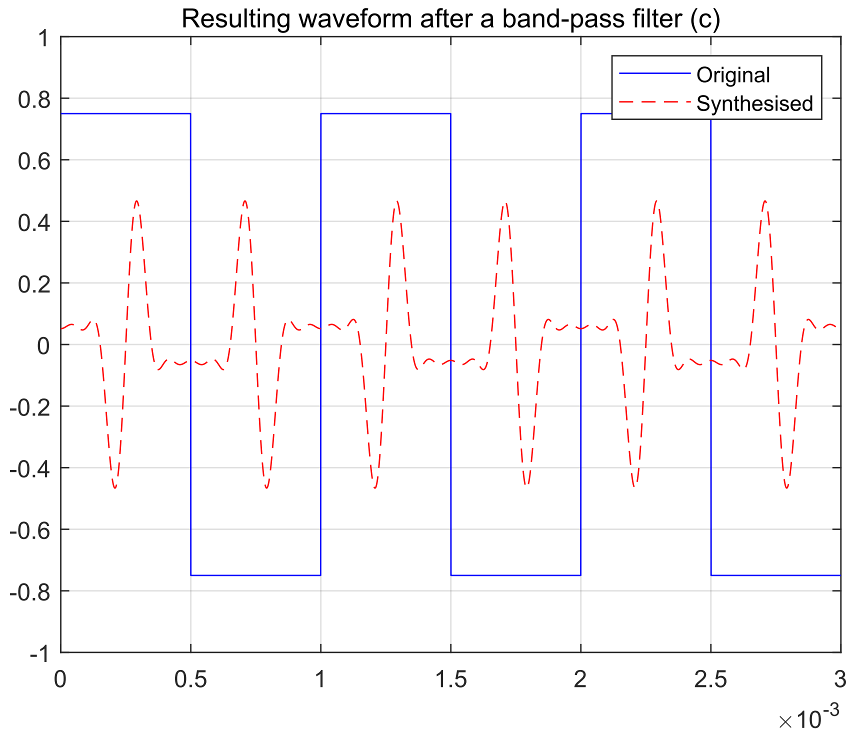

Figure 15. Resulting waveform after filter (c)

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

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

The fundamental frequency of the output from filter (c) is 1kHz. As shown in Fig.15, the period of synthesis waveform is 10^{-3}s, so the frequency is 1000Hz. And the fundamental frequency of synthesis waveform equals to the frequency of synthesis waveform itself. $f(n) = \frac{3}{\pi}[0 \cdot sin(\omega t - \pi) + \frac{1}{3} \cdot sin(3\omega t - \frac{3}{2}\pi) + \frac{1}{5} \cdot sin\,(5\omega t - \frac{5}{2}\pi) + \frac{1}{7} \cdot \sin\,(7\omega t - \frac{7}{2}\pi) + \frac{1}{9} \cdot \sin\,(9\omega t - \frac{9}{2}\pi), \omega = 2\pi f_0$. It can also be seen that the fundamental frequency of *f(n)* is f_0, which equals to 1kHz.

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]

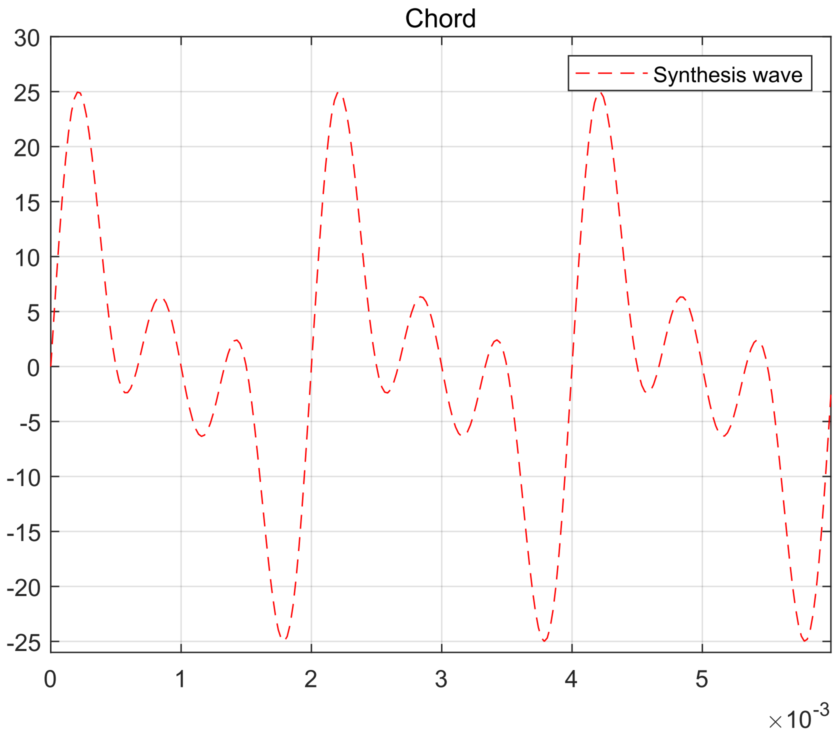

Figure 16. Waveform of chord (synthesis)

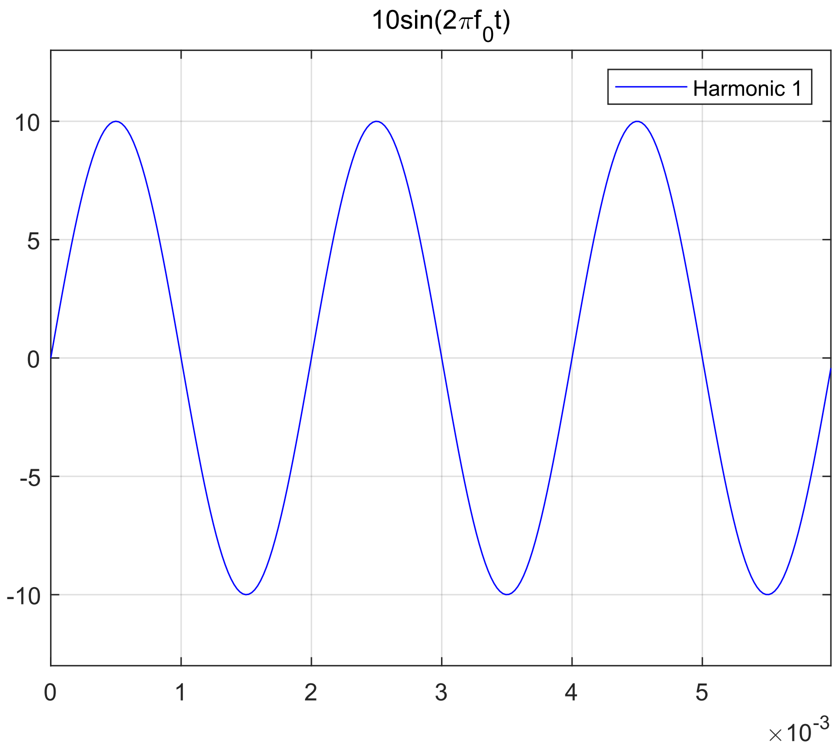

Figure 17. Waveform of harmonic 1

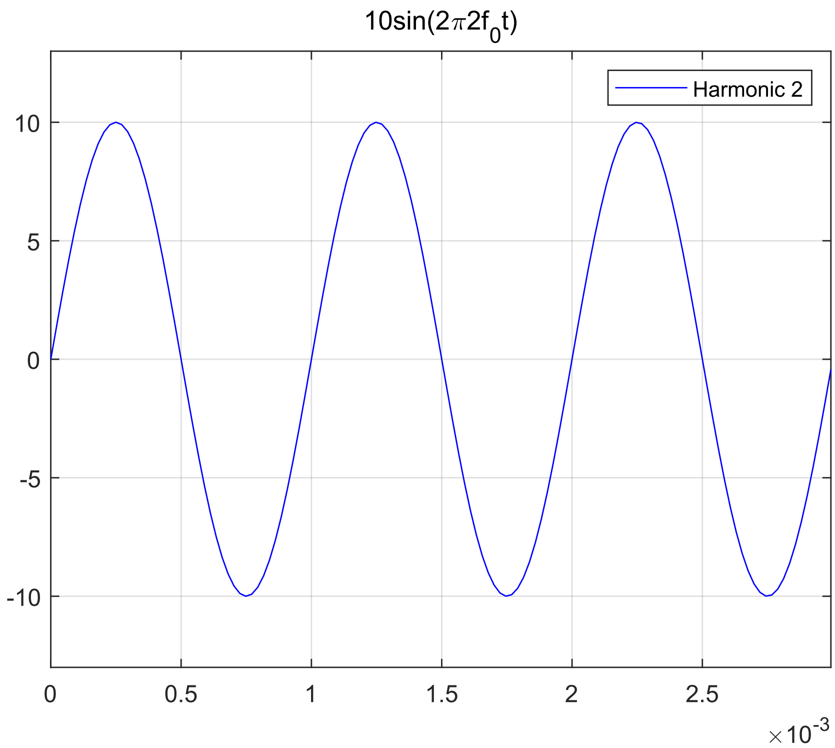

Figure 18. Waveform of harmonic 2

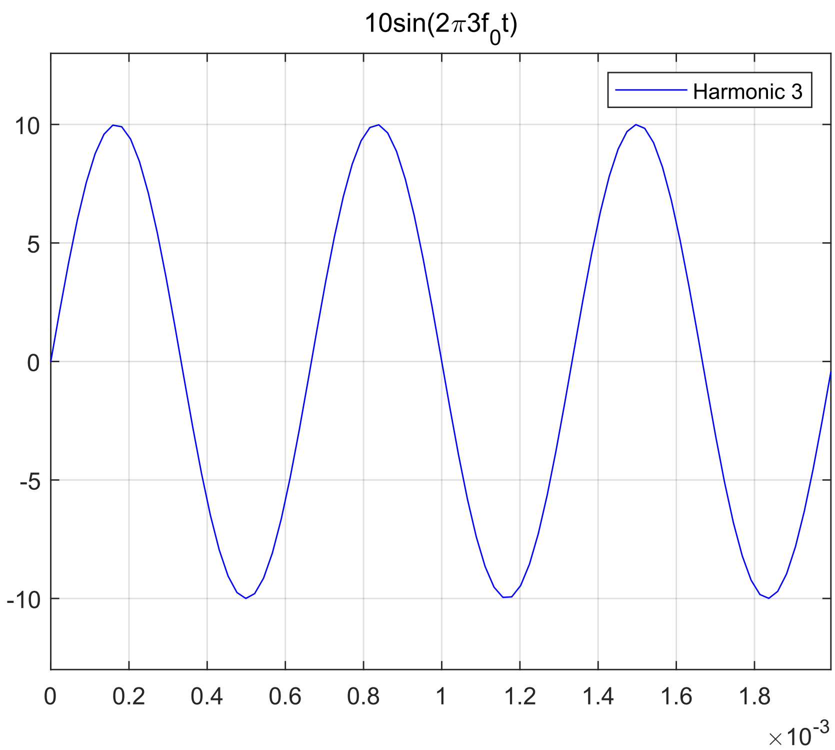

Figure 19. Waveform of harmonic 3

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

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

In this part, the third harmonic can be represented as: $10\sin(2\pi \cdot 3f_0 t+\frac{\pi}{2})$.

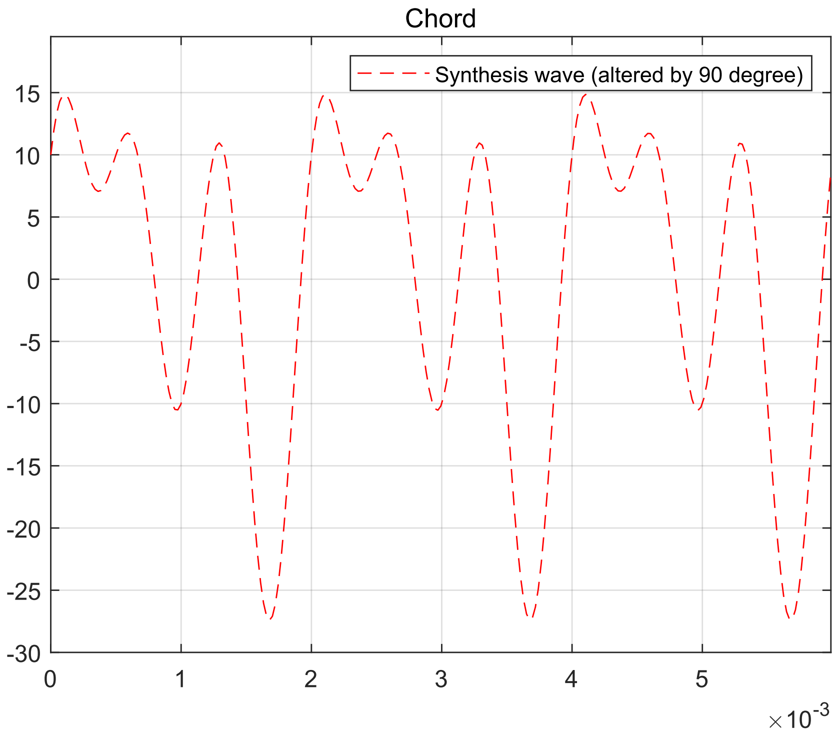

Figure 20. Waveform of chord (synthesis)

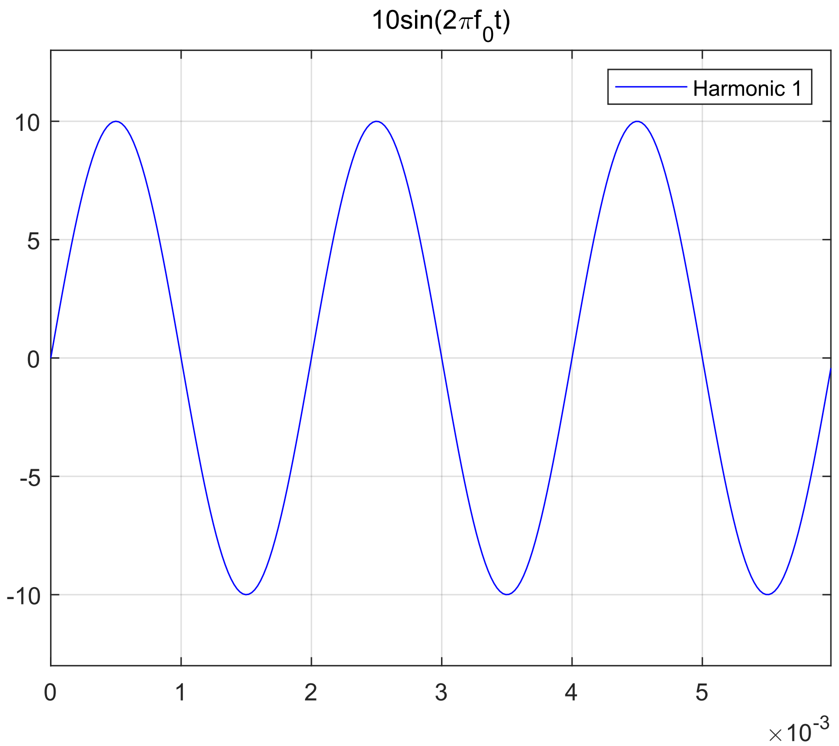

Figure 21. Waveform of harmonic 1

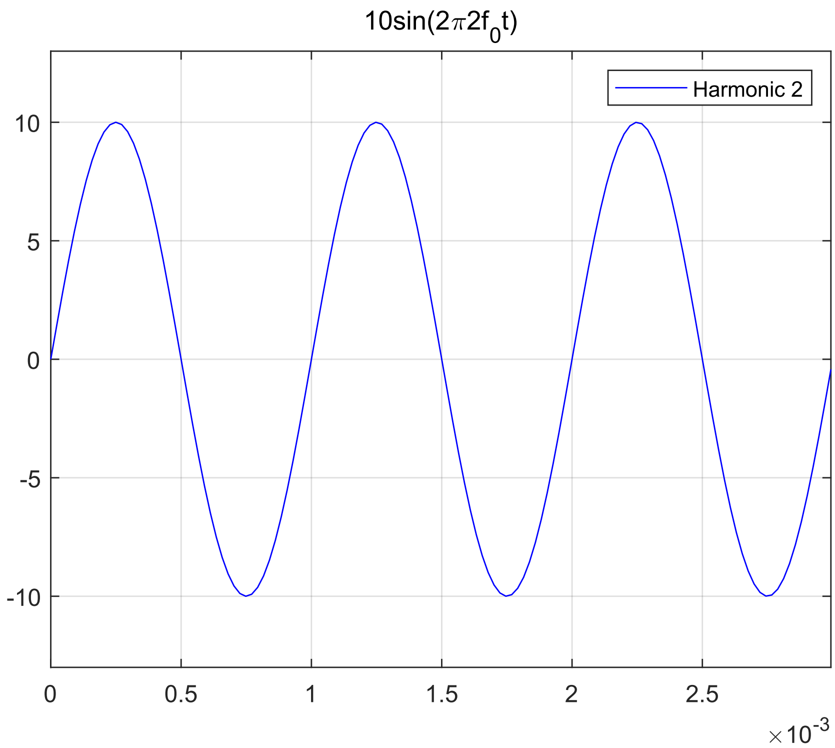

Figure 22. Waveform of harmonic 2

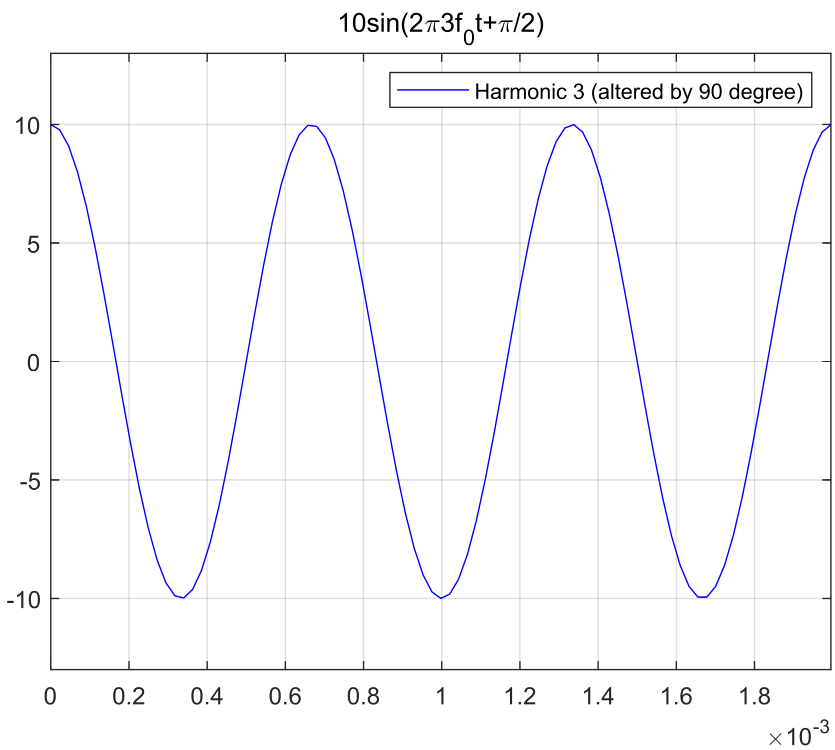

Figure 23. Waveform of harmonic 3 (altered by 90 degree)

MATLAB code are presented at Appendix part, as well as the screenshot of the whole program.

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]
When the phase of the third harmonic is altered, there is a difference in the plot of chords but the sound of two chords is consistent. For the plot, because the phase of the third harmonic

has changed, the image of the superposed waveform has also changed. For sound, the human ear is not sensitive to constant relative phase changes in a static waveform, so it is unable to distinguish the change in sound after a phase change.

E.4) What does the above tell you about the human ear? [2 marks]
The human ear can distinguish changes in frequency and amplitude in sound signals, but it is not sensitive to constant relative phase changes in a static waveform.

Appendix

Appendix.1) Source Code
A.1)

```
T = 1;
f0 = 1/T;
omega = 2*pi/T;
t = linspace(0, 3*T, 3*1000);
V0 = 3./pi;

synth_f =
V0.*sin(omega.*t)+1/3*sin(3.*omega.*t)+1/5*sin(5.*omega.*t)+1/7*sin(7.*omega.
*t)+1/9*sin(9.*omega.*t);

plot(t, synth_f,'r--')
legend('synthetic waveform')
axis([-inf inf -1.5 1.5]);
title('The synthetic waveform in Equation 12')
grid on
```

A.3) and A.6)

```
clear
% set signal period to 1s
T = 1;

% Time variable
t = linspace(0, 3.*T, 3000);

% Amplitute
V0 = 1;

% ------------------PartA Synthesis-----------------
max_harmonics = 9; % this number can be changed to 49, 99, 9999
a = zeros(1, max_harmonics+1);
b = zeros(1, max_harmonics+1);
synth_tri = zeros(size(t));

for n = 0 : max_harmonics
    if mod (n,2) == 1
        a(n+1) = V0./n;
        synth_tri = synth_tri + a(n+1)*sin(2*pi*n*(1/T)*t);
    end
end
plot(t, synth_tri,'r--')
legend('synthetic waveform')
axis([-inf inf -1.5 1.5]);
title('The synthetic waveform')
grid on
```

```
M = max(synth_tri, [], 'all');
N = min(synth_tri, [], 'all');
Amp = M - N;

%----------------A(6) Plot the spectrum--------------
figure
c = sqrt(a.^2 + b.^2);
stem(c)
title('The spectrum (frequency domain view) of f(t)')
grid on
```

B.3)

```
clc;clear
max_harmonics = 10;
T = 1;
V0 = 1;
b = zeros(1, max_harmonics+1);
t = linspace(0, 3*T, 3*1000);
synth_Sawtooth = zeros(size(t));

for n = 1 : max_harmonics
    b(n+1) = 2.*V0./((n).*pi);
    synth_Sawtooth = synth_Sawtooth + b(n+1)*sin(2*pi*n*(1/T)*t);
end

plot(t, synth_Sawtooth,'r--')
legend('synthetic waveform')
axis([-inf inf -1.5 1.5]);
grid on
title('The synthesised sawtooth waveforms (the number of harmonics is 10)')
set(gca, 'FontSize', 8)
```

B.4) and B.5)

```
function synth_Sawtooth = synsaw(max_harmonics)

% set signal period to 1s
T = 1;

% Time variable
t = linspace(0, T, 1000);

% Amplitute
V0 = 1;

% triangular function
Sawtooth = V0 - t.*2*V0./T;
t = linspace(0, 3*T, 3*1000);
Sawtooth = repmat (Sawtooth, 1, 3);
plot(t, Sawtooth,'b-')
hold on

% Synthesis
b = zeros(1, max_harmonics+1);
synth_Sawtooth = zeros(size(t));
```

```
for n = 1 : max_harmonics
    b(n+1) = 2.*V0./((n).*pi);
    synth_Sawtooth = synth_Sawtooth + b(n+1)*sin(2*pi*n*(1/T)*t);
end

plot(t, synth_Sawtooth,'r--')
legend('Sawtooth','synthetic waveform')
axis([-inf inf -1.5 1.5]);
grid on

end
```

```
clear
%---------n = 10-----------
synth_Sawtooth_10 = synsaw(10);
title('The original and synthesised sawtooth waveforms (the number of
harmonics is 10)')
set(gca, 'FontSize', 8)

%---------n = 5------------
figure
synth_Sawtooth_5 = synsaw(5);
title('The original and synthesised sawtooth waveforms (the number of
harmonics is 5)')
set(gca, 'FontSize', 8)
```

C.

```
clc; clear
% set signal period to 1s
T = 1;

% Time variable
t = linspace(0, 3*T, 3000);

% Amplitute
V0 = 1;

%----------room reservation-------
f1 = zeros(size(t));
f2 = zeros(size(t));
f3 = zeros(size(t));
f4 = zeros(size(t));

% ----------f1-----------
max_harmonics = 8;
a = zeros(1, max_harmonics+1);
for n = 0 : max_harmonics
    n1 = floor(n/2);
    if mod (n,2) == 1
        a(n) = ((-1).^n1)./n.^2;
        f1 = f1 + a(n)*sin(2*pi*n*(1/T)*t);
    end
end
figure
plot(t,f1,'r--')
grid on
```

```
title('The waveform of f1(t) synthesised by 4 harmonics')
legend('Synthesised f1(t)')

% ----------f2-----------
max_harmonics = 4;
b = zeros(1, max_harmonics+1);
for n = 1 : max_harmonics
    b(n) = 0.1;
    f2 = f2 + b(n)*cos(2*pi*n*(1/T)*t);
end
figure
plot(t, f2,'r--')
axis([-inf inf -0.2 0.5])
grid on
title('The waveform of f2(t) synthesised by 4 harmonics')
legend('Synthesised f2(t)')

% ----------f3-----------
max_harmonics = 10;
b = zeros(1, max_harmonics+1);
for n = 1 : max_harmonics
    if mod (n,2) == 0
        b(n) = - 4 / ((n^2-1) * pi);
        f3 = f3 + b(n) * cos(2 * pi * n * (1/T) * t);
    end
end
f3 = sin(2 * pi * (1/T) * t) +f3;
figure
plot(t,f3,'r--')
axis([-inf inf -1 1.65])
grid on
title('The waveform of f3(t) synthesised by 6 harmonics')
legend('Synthesised f3(t)')

% ----------f4-----------
max_harmonics = 10;
b = zeros(1, max_harmonics+1);
for n = 1 : max_harmonics
    if mod (n,2) == 0
        b(n) = - 4./((n^2-1) * pi);
        f4 = f4 + b(n)*cos(2 * pi * n * (1/T) * t);
    end
end
figure
plot(t,f4,'r--')
axis([-inf inf -0.7 0.5])
grid on
title('The waveform of f4(t) synthesised by 6 harmonics')
legend('Synthesised f4(t)')
```

D.

```
function synth_sqr = synsqr(max_harmonics)

% set signal frequency to 1000Hz
f0 = 1000;  % (1 khz)

% Time variable
```

```
t = linspace(0, 1./f0, 1000);

% Amplitute
V0 = 3./pi;

% ----------synth sqr----------

a = zeros(1, max_harmonics+1);
b = zeros(1, max_harmonics+1);
synth_sqr = zeros(size(t));


for n = 0 : max_harmonics
    if mod (n,2) == 1
        a(n) = V0./n;
        synth_sqr = synth_sqr + a(n)*sin(2*pi*n*f0*t);
    end
end

end
```

```
clc; clear

% -----------Synthesis(D.1)--------------
f0 = 1000;
t = linspace(0, 1./f0, 1000);
sqr1 (t<=1./(2.*f0)) = 3/4;  % Caution: this 3/4 should be calculated
precisely !
sqr1 (t>1./(2.*f0))= -3/4;
synth_sqr_99 = synsqr(99);
figure
plot(synth_sqr_99,'r--')
hold on
plot(sqr1,'b-')
grid on
title('Synthesis and original waveform of V(t)')
legend('Synthesised','Original')


%-----------overshoot------------
M = max(synth_sqr_99, [], 'all');
overshoot = M - 3/4;
percentage_overshoot = overshoot./(3/4);
sprintf('Percentage overshoot is %4.3f%%',percentage_overshoot*100)


%---------------3(a)----------------
max_harmonics = 10;
f0 = 1000;
t = linspace(0, 3./f0, 3000);
a = zeros(1, max_harmonics+1);
b = zeros(1, max_harmonics+1);
synth_f1_lpf = zeros(size(t));
synth_f2_lpf = zeros(size(t));
V0 = 3./pi;

% ---sqr function---
t = linspace(0, 1./f0, 1000);
```

```
sqr (t<=1./(2.*f0)) = 3/4;  % Caution: this 3/4 should be calculated
precisely !
sqr (t>1./(2.*f0))= -3/4;


t = linspace(0, 3./f0, 3000);
sqr = repmat (sqr, 1, 3);


for n = 0 : max_harmonics
    if mod (n,2) == 1
        a(n) = V0./n;
        synth_f1_lpf = synth_f1_lpf + a(n)*sin(2*pi*n*f0*t-pi.*n./2);
    end
end


%---plot---
figure
plot(t, sqr,'b-')
hold on
plot(t, synth_f1_lpf,'r--')
axis([-inf inf -1.5 1.5])
grid on
title('Resulting waveform after a low-pass filter (a)')
legend('Original','Synthesised')


%---------------3(b)-----------------
for n = 0 : max_harmonics
    if mod (n,2) == 1
        a(n) = V0./n;
        synth_f2_lpf = synth_f2_lpf + a(n)*sin(2*pi*n*f0*t-pi./2);
    end
end

%---plot---
figure
plot(t, sqr,'b-')
hold on
plot(t, synth_f2_lpf,'r--')
axis([-inf inf -2.2 2.2])
grid on
title('Resulting waveform after a low-pass filter (b)')
legend('Original','Synthesised')




%---------------3(c)-----------------
f0 = 1000;
t = linspace(0, 3./f0, 3*1000);
V0 = 3./pi;

synth_f3_lpf = V0.*(0.*sin(2.*pi.*f0.*t-
0.5.*2*pi)+1./3.*0.5.*sin(2.*pi.*3.*f0.*t-
1.5*pi)+1./5.*1.*sin(2.*pi.*5.*f0.*t-2.5*pi)+1./7.*1.*sin(2.*pi.*7.*f0.*t-
3.5*pi)+1./9.*0.5.*sin(2.*pi.*9.*f0.*t-4.5*pi));
figure
```

```
plot(t, sqr,'b-')
hold on
plot(t, synth_f3_lpf,'r--')
axis([-inf inf -1.0 1.0])
grid on
title('Resulting waveform after a band-pass filter (c)')
legend('Original','Synthesised')
```

E.1)

```
clc; clear

f = 500;
fs = 44100;
duration = 3;
t = 0:1/fs:duration;
A = 10; % amplitude
harmonic_1 = A*sin(2*pi*f*t);
harmonic_2 = A*sin(2*pi*2*f*t);
harmonic_3 = A*sin(2*pi*3*f*t);
wave_synth = harmonic_1 + harmonic_2 + harmonic_3;
sound(wave_synth,fs)
T_max = 3;
clipped_t = t(1:find(t<=T_max/f,1,'last'));

clipped_wave_synth = wave_synth(1:find(t<=T_max/f,1,'last'));

clipped_harmonic_1 = harmonic_1(1:find(t<=T_max/f,1,'last'));

clipped_t_2 = t(1:find(t<=T_max/(2*f),1,'last'));
clipped_harmonic_2 = harmonic_2(1:find(t<=T_max/(2*f),1,'last'));

clipped_t_3 = t(1:find(t<=T_max/(3*f),1,'last'));
clipped_harmonic_3 = harmonic_3(1:find(t<=T_max/(3*f),1,'last'));


figure
plot(clipped_t, clipped_wave_synth,'r--')
legend('Synthesis wave')
grid on
title('Chord')
axis([-inf inf -26 30]);

figure
plot(clipped_t, clipped_harmonic_1,'b-')
legend('Harmonic 1')
grid on
title('10sin(2\pif_0t)')
axis([-inf inf -13 13]);

figure
plot(clipped_t_2, clipped_harmonic_2,'b-')
legend('Harmonic 2')
grid on
title('10sin(2\pi2f_0t)')
axis([-inf inf -13 13]);

figure
```

```
plot(clipped_t_3, clipped_harmonic_3,'b-')
legend('Harmonic 3')
grid on
title('10sin(2\pi3f_0t)')
axis([-inf inf -13 13]);
```

E.2)

```
clc; clear
f = 500;
fs = 44100;
duration = 3;
t = 0:1/fs:duration;
A = 10; % amplitude
phai = pi/2;
harmonic_1 = A*sin(2*pi*f*t);
harmonic_2 = A*sin(2*pi*2*f*t);
harmonic_3 = A*sin(2*pi*3*f*t-phai);
wave_synth = harmonic_1 + harmonic_2 + harmonic_3;
sound(wave_synth,fs)
T_max = 3;
clipped_t = t(1:find(t<=T_max/f,1,'last'));
clipped_wave_synth = wave_synth(1:find(t<=T_max/f,1,'last'));
clipped_harmonic_1 = harmonic_1(1:find(t<=T_max/f,1,'last'));

clipped_t_2 = t(1:find(t<=T_max/(2*f),1,'last'));
clipped_harmonic_2 = harmonic_2(1:find(t<=T_max/(2*f),1,'last'));

clipped_t_3 = t(1:find(t<=T_max/(3*f),1,'last'));
clipped_harmonic_3 = harmonic_3(1:find(t<=T_max/(3*f),1,'last'));


figure
plot(clipped_t, clipped_wave_synth,'b-')
legend('Synthesis wave (altered by 90 degree)')
grid on
axis([-inf inf -18 33]);
figure
plot(clipped_t, clipped_harmonic_1,'b-')
legend('Harmonic 1')
grid on
axis([-inf inf -13 13]);
figure
plot(clipped_t_2, clipped_harmonic_2,'b-')
legend('Harmonic 2')
grid on
axis([-inf inf -13 13]);
figure
plot(clipped_t_3, clipped_harmonic_3,'b-')
legend('Harmonic 3 (altered by 90 degree)')
grid on
axis([-inf inf -13 13]);
```

Appendix.2) Screenshot

A.1)

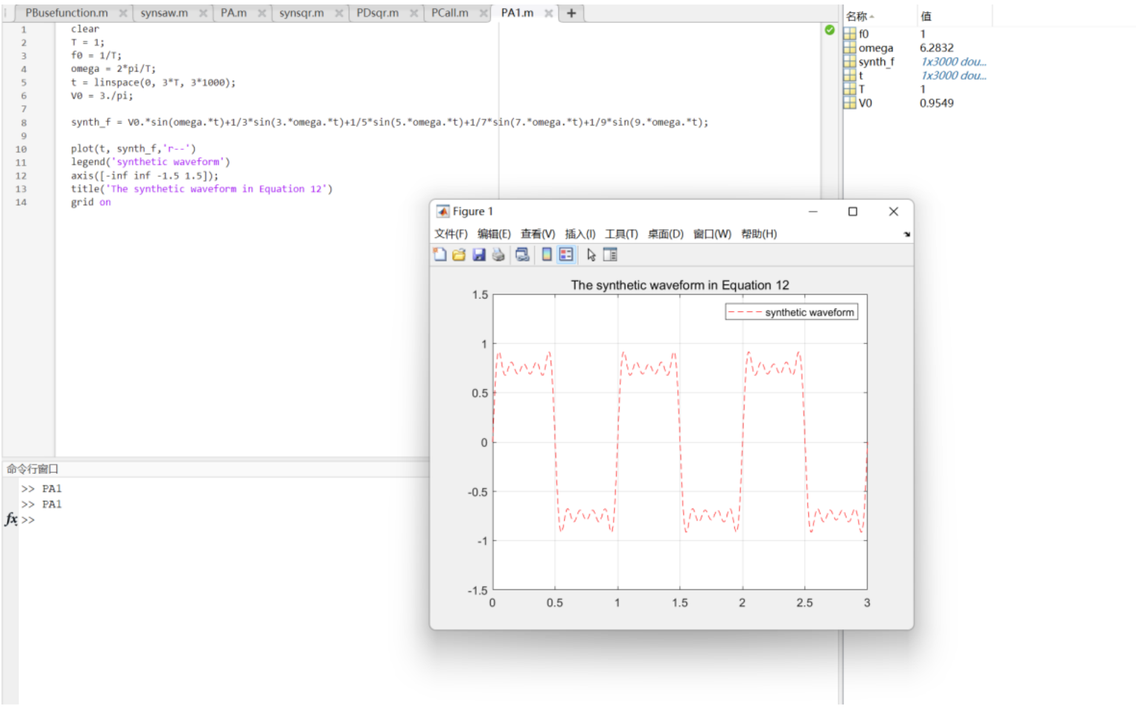

Figure 24. Screenshot of Part A.1

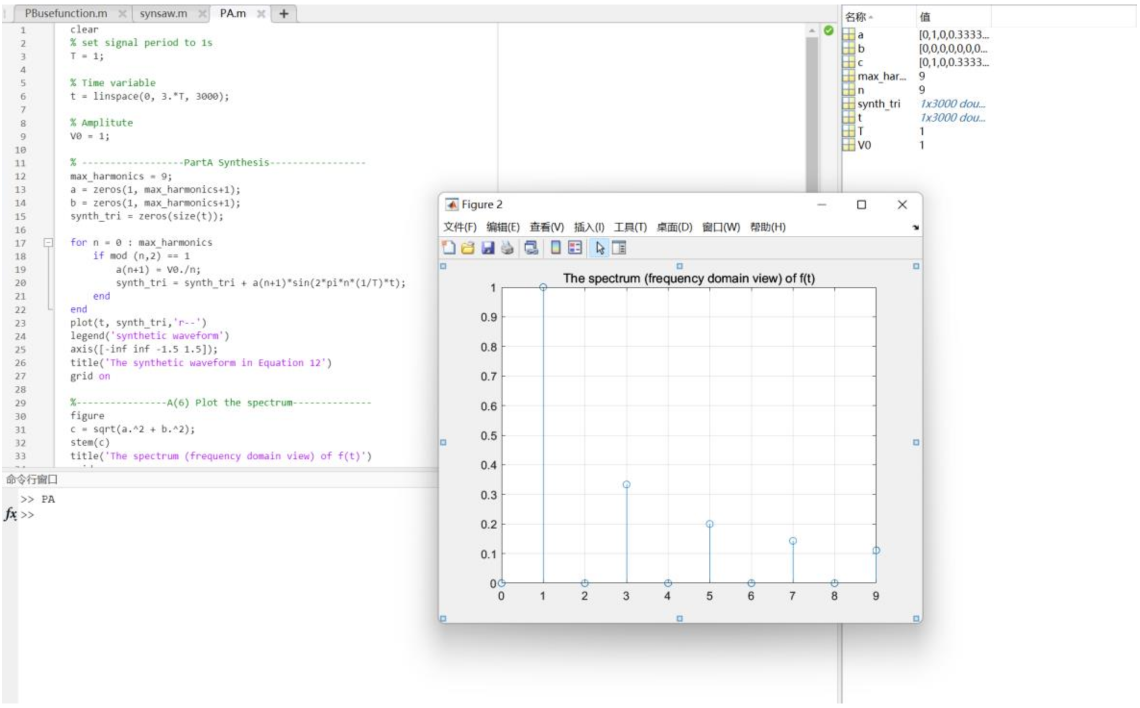

Figure 25. Screenshot of Part A.2

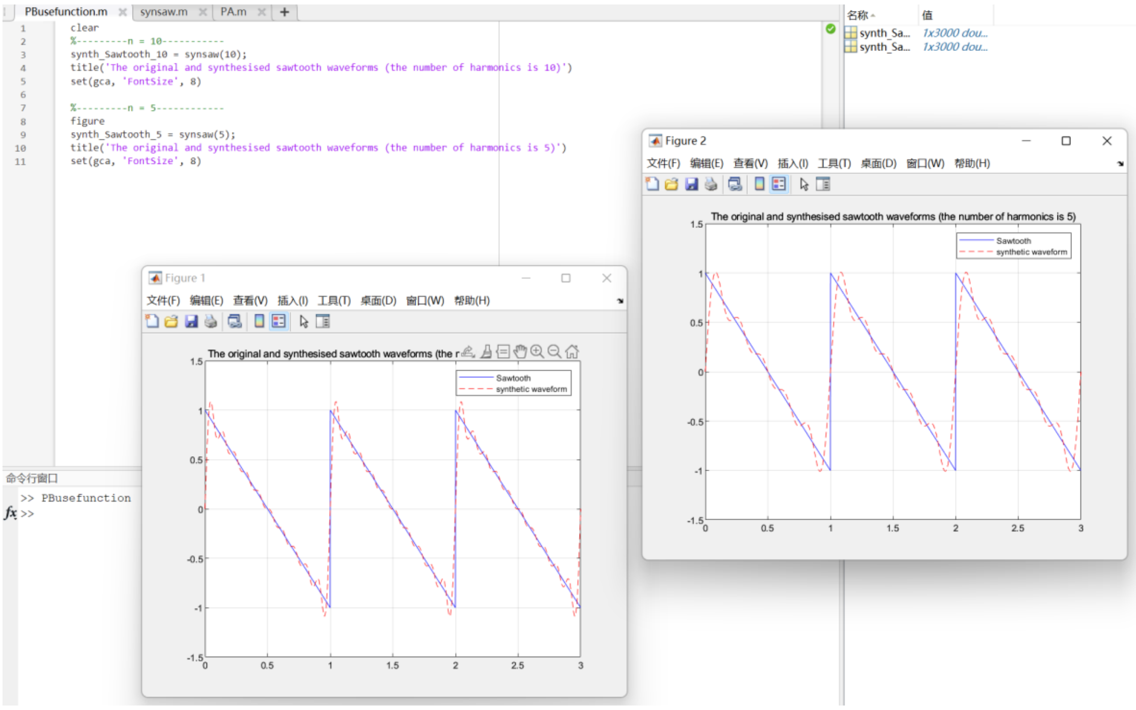

Figure 26. Screenshot of Part B

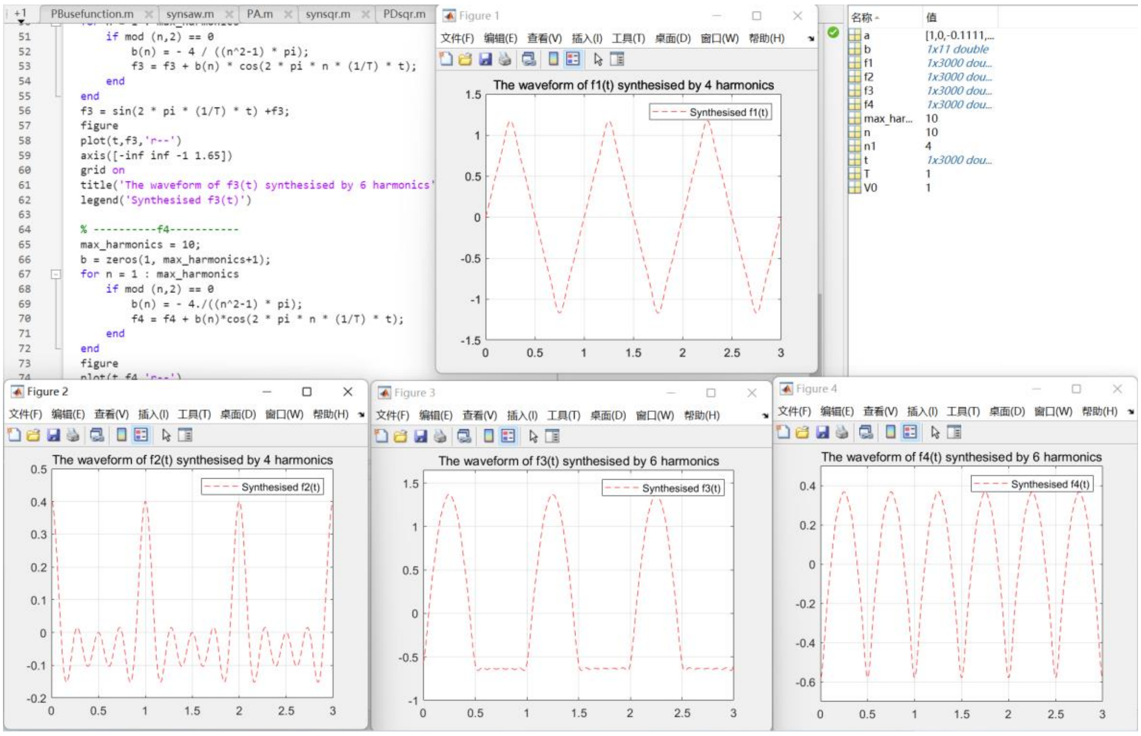

Figure 27. Screenshot of Part C

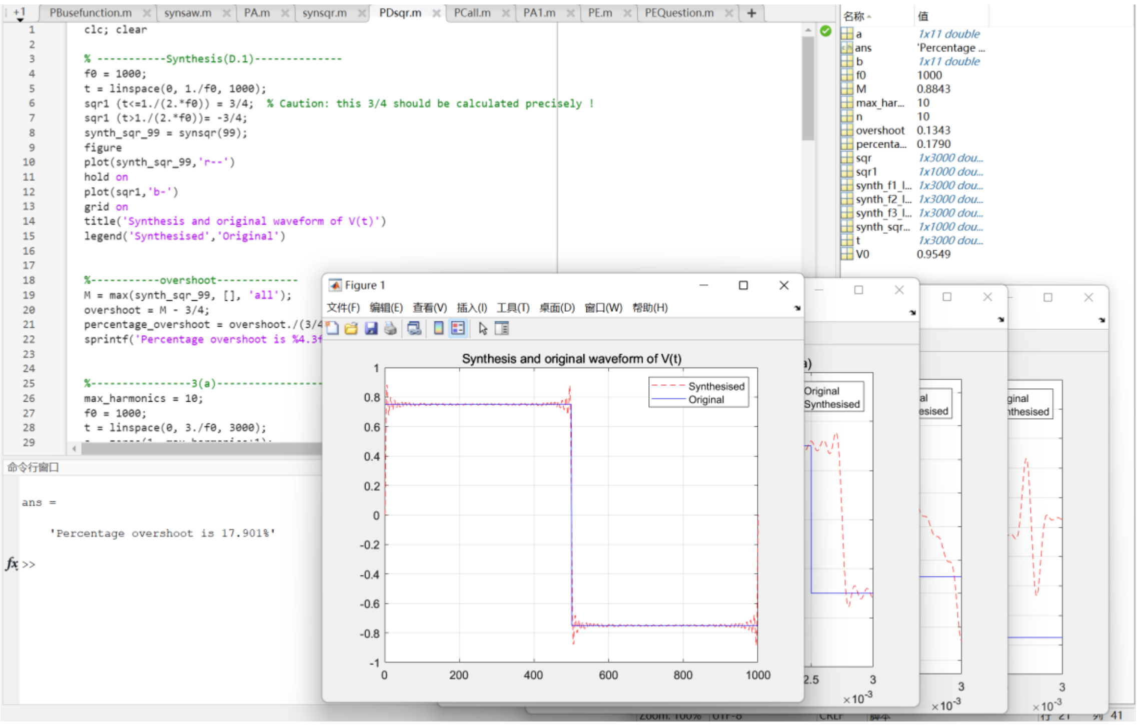

Figure 28. Screenshot of Part D

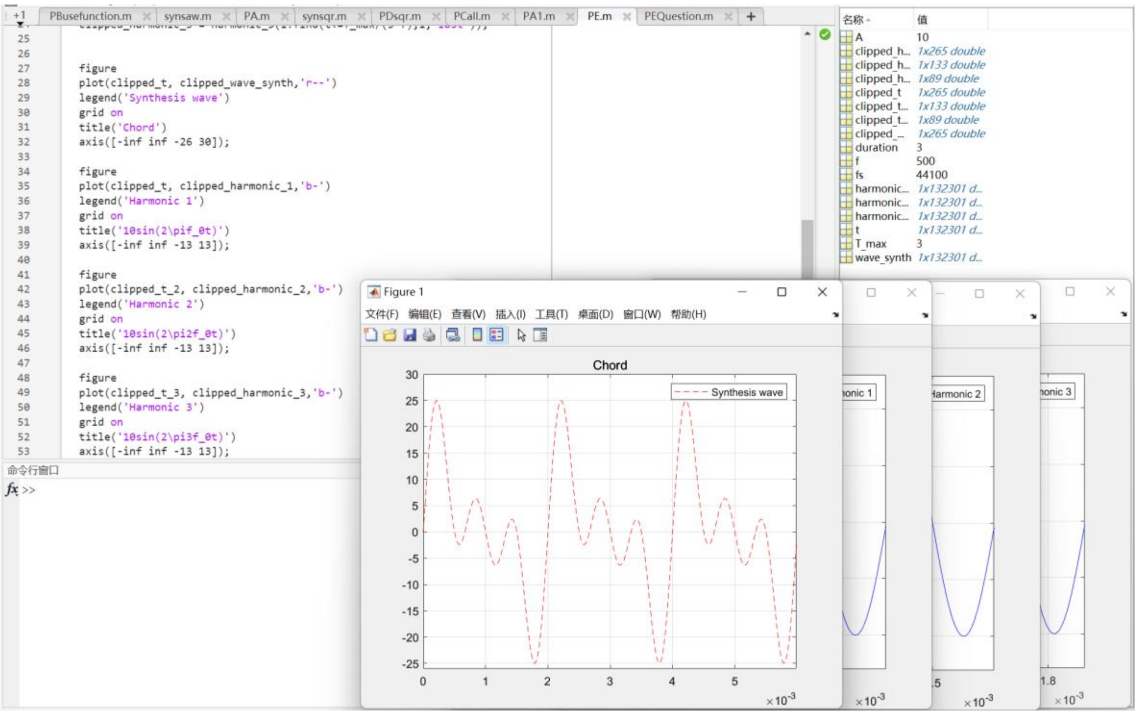

Figure 29. Screenshot of Part E.1

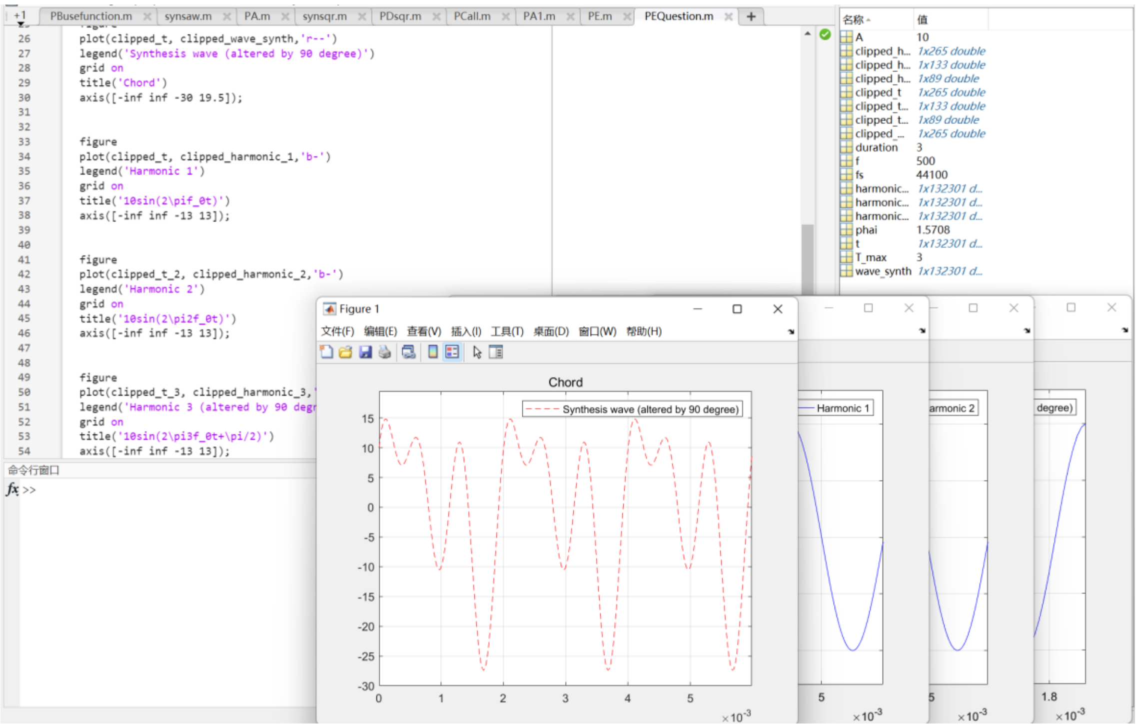

Figure 30. Screenshot of Part E.2