Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Number	ELE532
Course Title	Signals and Systems I
Semester/Year	F2025
Instructor	Soosan Beheshti
TA Name	Dhanisha Trivedi

Lab/Tutorial Report No.	3
Report Title	Fourier Series Analysis using Matlab

Submission Date	November, 7th 2025
Due Date	November, 8th 2025

Student Name	Student ID	Signature
Dian Zahid	501253063	D.Z
Hamzah Sahi	501221955	H.S

Introduction

The purpose of this lab is to use MATLAB to analyze periodic signals through Fourier series. The experiment involves calculating Fourier coefficients, plotting magnitude and phase spectra, and reconstructing signals from a limited number of harmonics. It also examines how a signal's period and the number of coefficients affect the accuracy of reconstruction.

Part A)

A.1

A_1: $x_1(t) = \cos \frac{3\pi}{10} t + \frac{1}{2} \cos \frac{\pi}{10} t \qquad W_0 = \frac{\pi}{10} \quad T_0 = \frac{2\pi}{W_0} = 20$

$\cos(k w_0 t) = \frac{1}{2}(e^{jkw_0 t} + e^{-jkw_0 t})$

$x_1(t) = \frac{1}{2}(e^{j3w_0 t} + e^{-j3w_0 t}) + \frac{1}{4}(e^{jw_0 t} + e^{-jw_0 t}) = \sum_{n=-\infty}^{\infty} D_n e^{jnw_0 t}$

$\therefore D_{\pm 3} = \frac{1}{2}, \quad D_{\pm 1} = \frac{1}{4}, \quad D_n = 0$ otherwise

A.2

A_2. $D_n = \frac{1}{T_0} \int_{t_0}^{t_0 + T_0} x(t) e^{-jnw_0 t} \, dt \qquad W_0 = \frac{2\pi}{T_0}$

$D_n = \frac{A}{T_0} = \int_{ts}^{ts+\tau} e^{-jnw_0 t} \, dt = \frac{A}{T_0} \frac{1 - e^{-jnw_0 \tau}}{jnw_0} e^{-jnw_0 ts}$

$D_0 = \frac{A}{T_0} \int_{ts}^{ts+\tau} 1 \, dt = \frac{A\tau}{T_0} \qquad D_n = \frac{A\tau}{T_0} \operatorname{sinc}\left(\frac{nW_0\tau}{2}\right) e^{-jnw_0(ts + \tau/2)} \quad \left\{ \operatorname{sinc}(x) = \frac{\sin x}{x} \right.$

$x_2(t)$: $A = 1, \ \tau = 10, T_0 = 20 \quad W_0 = \pi/10$

$D_n = \frac{\tau}{T_0} \operatorname{sinc}\left(\frac{nW_0\tau}{2}\right) = \frac{10}{20} \operatorname{sinc}\left(\frac{n(\pi/10) \cdot 10}{2}\right)$

$D_n = \frac{1}{2} \operatorname{sinc}\left(\frac{n\pi}{2}\right)$

$x_3(t)$: $A = 1, \ \tau = 10, T_0 = 40 \quad W_0 = \pi/20$

$D_n = \frac{\tau}{T_0} \operatorname{sinc}\left(\frac{nW_0\tau}{2}\right) = \frac{10}{40} \operatorname{sinc}\left(\frac{n(\pi/20) \cdot 10}{2}\right)$

$D_n = \frac{1}{4} \operatorname{sinc}\left(\frac{n\pi}{4}\right)$

A.3

Dn_x1.m × Dn_rect.m × A4.m ×

/Users/hamzah/Documents/MATLAB/Dn_x1.m

```matlab
function D = Dn_x1(nvec)
    D = zeros(size(nvec)); % preallocate
    for k = 1:numel(nvec)
        n = nvec(k);
        if abs(n) == 3
            D(k) = 1/2;       % from cos(3ω0 t) = 0.5(e^{j3ω0t}+e^{-j3ω0t})
        elseif abs(n) == 1
            D(k) = 1/4;       % from 0.5 cos(ω0 t) = 0.25(e^{jω0t}+e^{-jω0t})
        else
            D(k) = 0;
        end
    end
end
```

Figure 1: Dn X1 code

/Users/hamzah/Documents/MATLAB/Dn_rect.m

```matlab
function D = Dn_rect(nvec, A, T0, tau, ts)
    w0 = 2*pi/T0;
    D  = zeros(size(nvec));
    for k = 1:numel(nvec)
        n = nvec(k);
        if n == 0
            D(k) = A * (tau / T0);
        else
            D(k) = (A/T0) * (1 - exp(-1j*n*w0*tau)) / (1j*n*w0) * exp(-1j*n*w0*ts);
        end
    end
end
```

Figure 2: Dn X2/X3 code

A.4

/Users/hamzah/Documents/MATLAB/A4.m

```matlab
% Parameters
A_x2 = 1; T0_x2 = 20; tau_x2 = 10; ts_x2 = -5;
A_x3 = 1; T0_x3 = 40; tau_x3 = 10; ts_x3 = -5;

ranges = { -5:5, -20:20, -50:50, -500:500 };
labels = { 'n=-5..5','n=-20..20','n=-50..50','n=-500..500' };

% x1
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_x1(nvec);
    figure;
    subplot(2,1,1); stem(nvec,abs(D),'filled'); grid on
    ylabel('|D_n|'); title(['x_1  ', labels{i}])
    subplot(2,1,2); stem(nvec,angle(D),'filled'); grid on
    xlabel('n'); ylabel('\angle D_n (rad)')
end

% x2
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_rect(nvec, A_x2, T0_x2, tau_x2, ts_x2);
    figure;
    subplot(2,1,1); stem(nvec,abs(D),'filled'); grid on
    ylabel('|D_n|'); title(['x_2  ', labels{i}])
    subplot(2,1,2); stem(nvec,angle(D),'filled'); grid on
    xlabel('n'); ylabel('\angle D_n (rad)')
end

% x3
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_rect(nvec, A_x3, T0_x3, tau_x3, ts_x3);
    figure;
    subplot(2,1,1); stem(nvec,abs(D),'filled'); grid on
    ylabel('|D_n|'); title(['x_3  ', labels{i}])
    subplot(2,1,2); stem(nvec,angle(D),'filled'); grid on
    xlabel('n'); ylabel('\angle D_n (rad)')
end
```

Figure 3: A4 Code

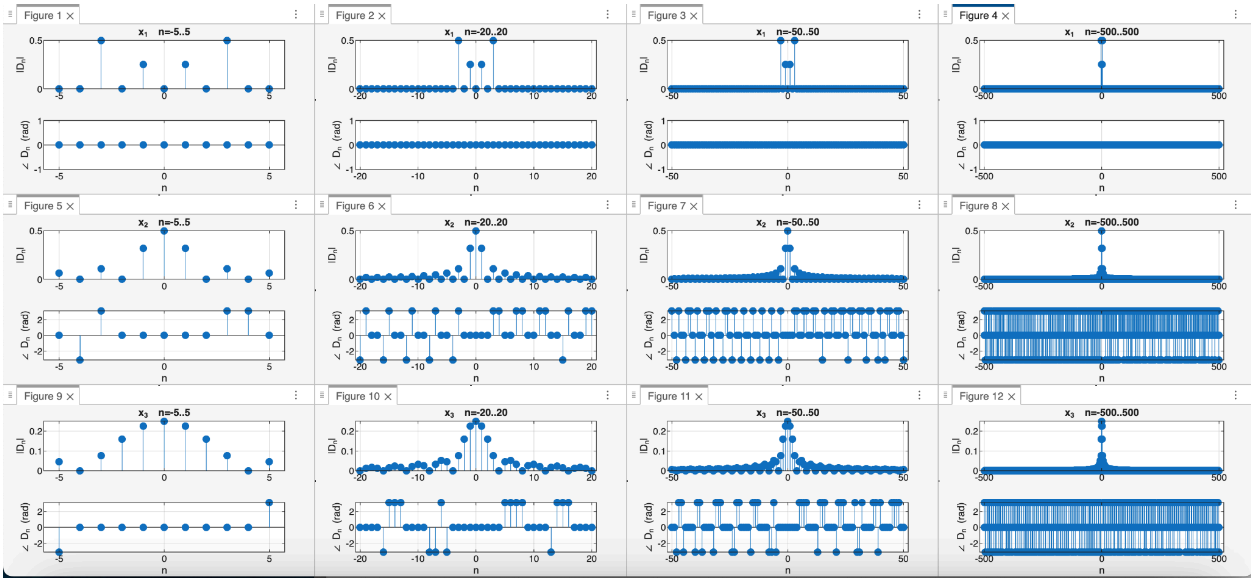


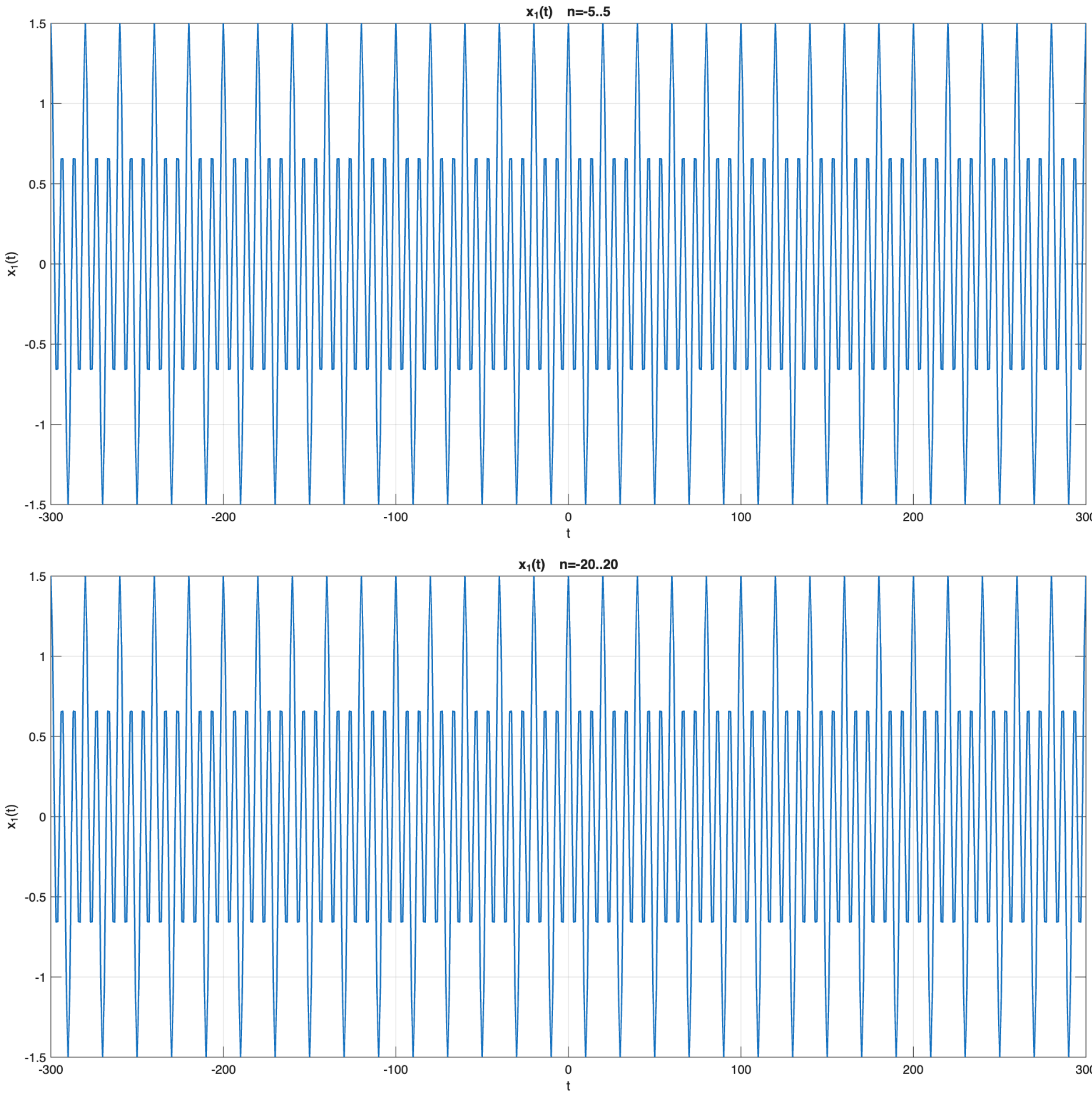
Figure 4: X1(t), X2(t), X3(t) magnitude and phase spectra plots

A.5/A.6

/Users/hamzah/Documents/MATLAB/A5_A6.m

```matlab
% A5/A6

% Parameters
T0_x1 = 20;
A_x2 = 1; T0_x2 = 20; tau_x2 = 10; ts_x2 = -5;
A_x3 = 1; T0_x3 = 40; tau_x3 = 10; ts_x3 = -5;

ranges = { -5:5, -20:20, -50:50, -500:500 };
labels = { 'n=-5..5','n=-20..20','n=-50..50','n=-500..500' };

% x1 reconstructions
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_x1(nvec);
    [t,xrec] = reconstruct_from_Dn(nvec, D, T0_x1);
    figure; plot(t, xrec, 'LineWidth', 1.2); grid on
    title(['x_1(t)  ', labels{i}]); xlabel('t'); ylabel('x_1(t)')
end

% x2 reconstructions
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_rect(nvec, A_x2, T0_x2, tau_x2, ts_x2);
    [t,xrec] = reconstruct_from_Dn(nvec, D, T0_x2);
    figure; plot(t, xrec, 'LineWidth', 1.2); grid on
    title(['x_2(t)  ', labels{i}]); xlabel('t'); ylabel('x_2(t)')
end

% x3 reconstructions
for i = 1:numel(ranges)
    nvec = ranges{i};
    D = Dn_rect(nvec, A_x3, T0_x3, tau_x3, ts_x3);
    [t,xrec] = reconstruct_from_Dn(nvec, D, T0_x3);
    figure; plot(t, xrec, 'LineWidth', 1.2); grid on
    title(['x_3(t)  ', labels{i}]); xlabel('t'); ylabel('x_3(t)')
end

% local functions

function [t, xrec] = reconstruct_from_Dn(nvec, D, T0)
    w0 = 2*pi/T0;
    t = -300:300;
    xrec = zeros(size(t));
    for k = 1:numel(nvec)
        n = nvec(k);
        xrec = xrec + D(k).*exp(1j*n*w0*t);
    end
    xrec = real(xrec);
end

function D = Dn_x1(nvec)
    D = zeros(size(nvec));
    for k = 1:numel(nvec)
        n = nvec(k);
        if abs(n)==3, D(k)=1/2;
        elseif abs(n)==1, D(k)=1/4;
        else, D(k)=0;
        end
    end
end

function D = Dn_rect(nvec, A, T0, tau, ts)
    w0 = 2*pi/T0; D=zeros(size(nvec));
    for k=1:numel(nvec)
        n=nvec(k);
        if n==0
            D(k)=A*(tau/T0);
        else
            D(k)=(A/T0)*(1-exp(-1j*n*w0*tau))/(1j*n*w0)*exp(-1j*n*w0*ts);
        end
    end
end
```

Figure 5: A5/A6 Code



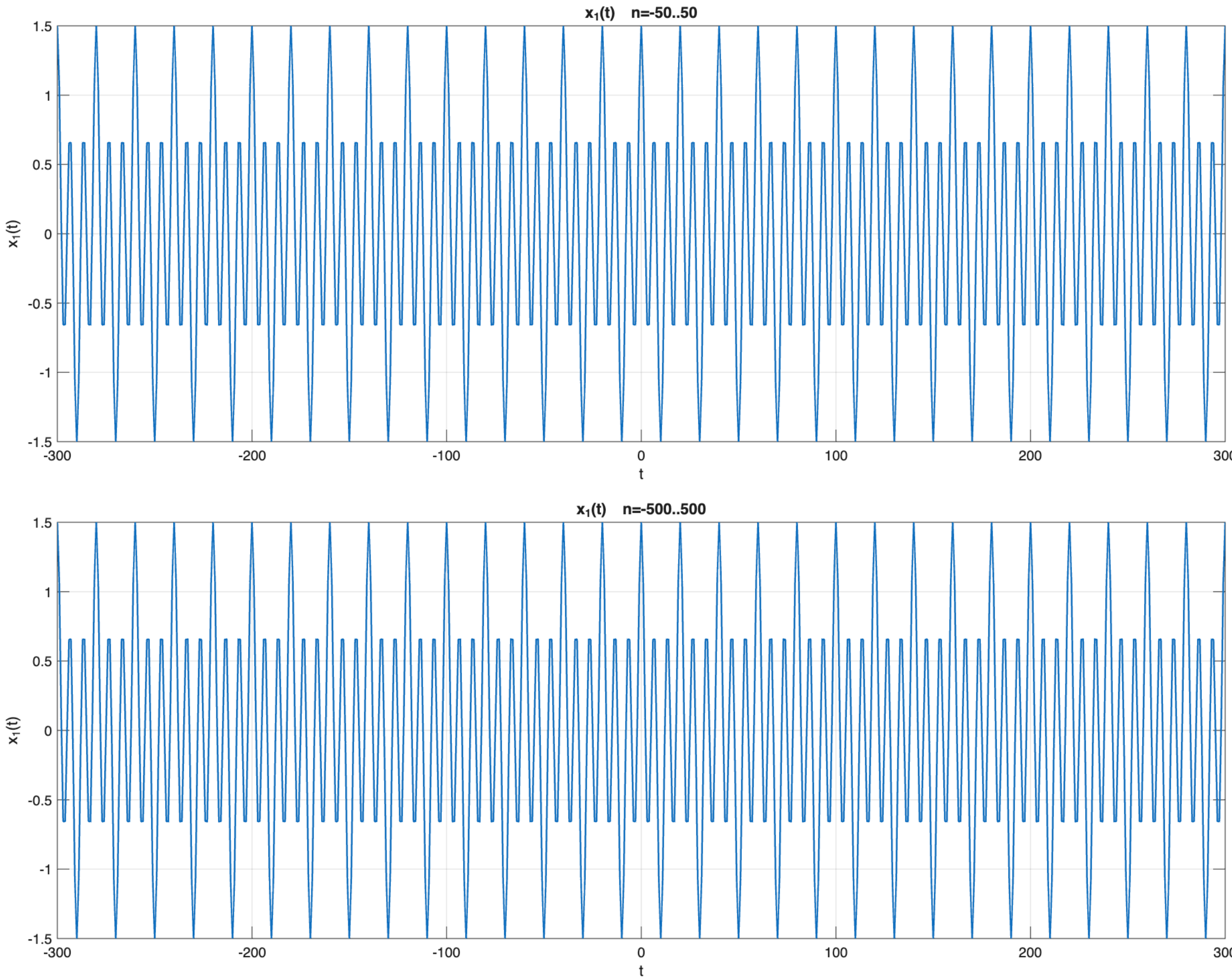
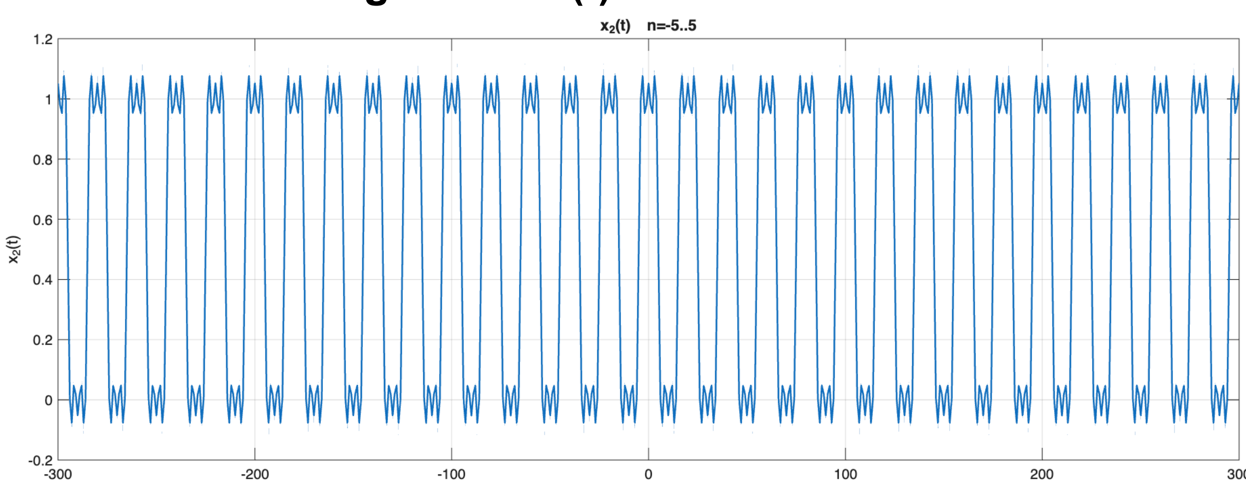


Figure 6: X1(t) Reconstructed



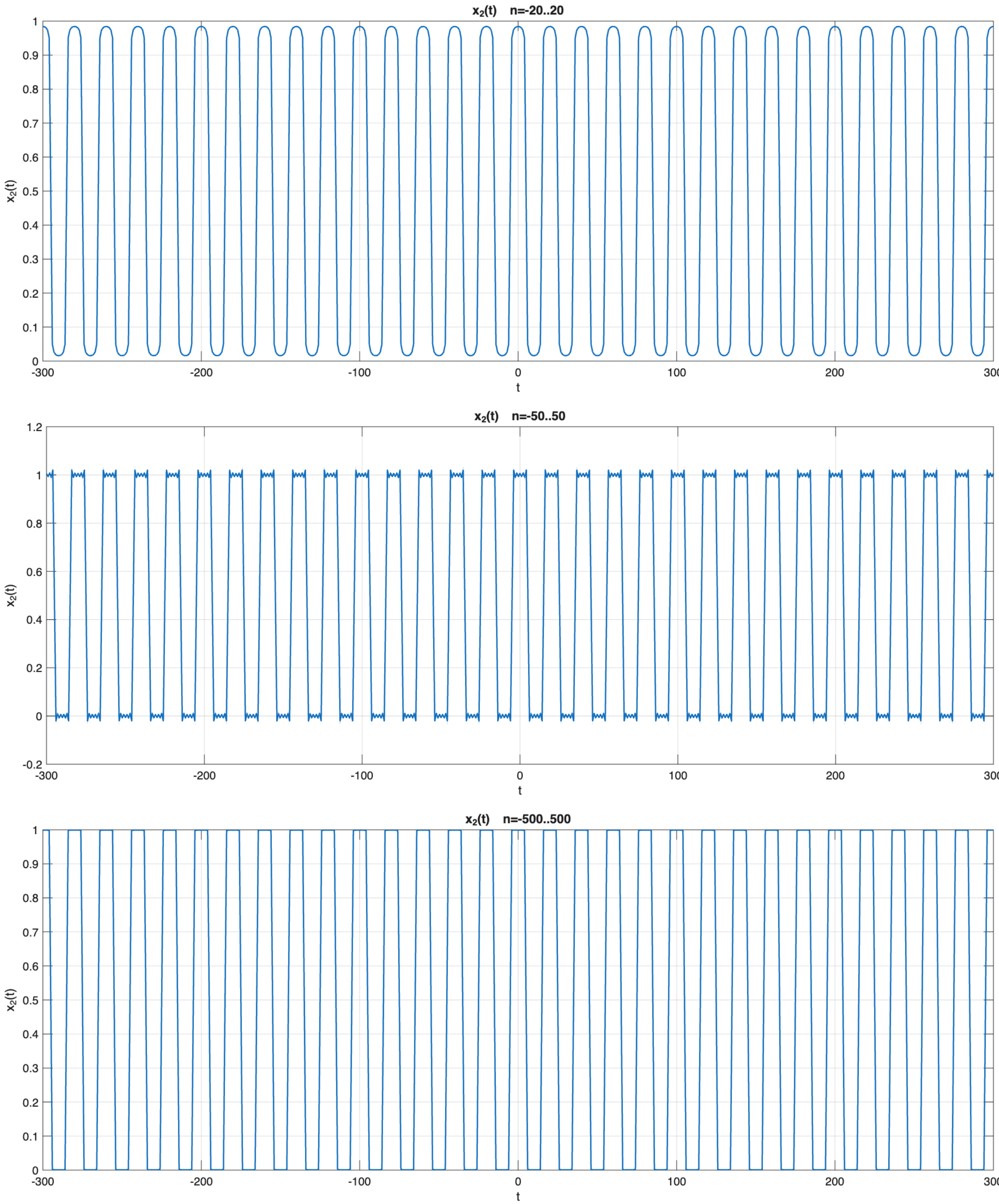
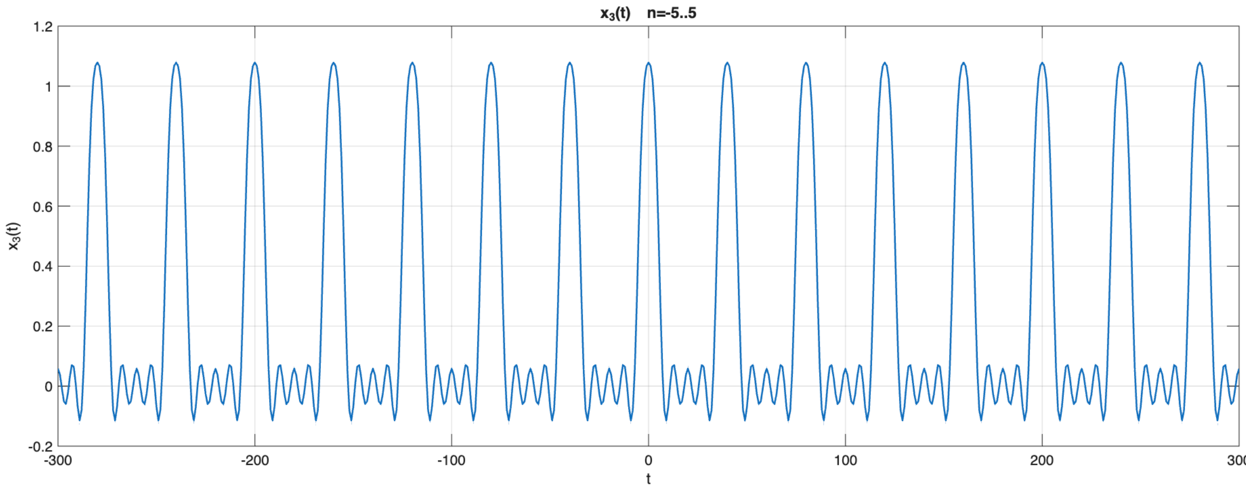
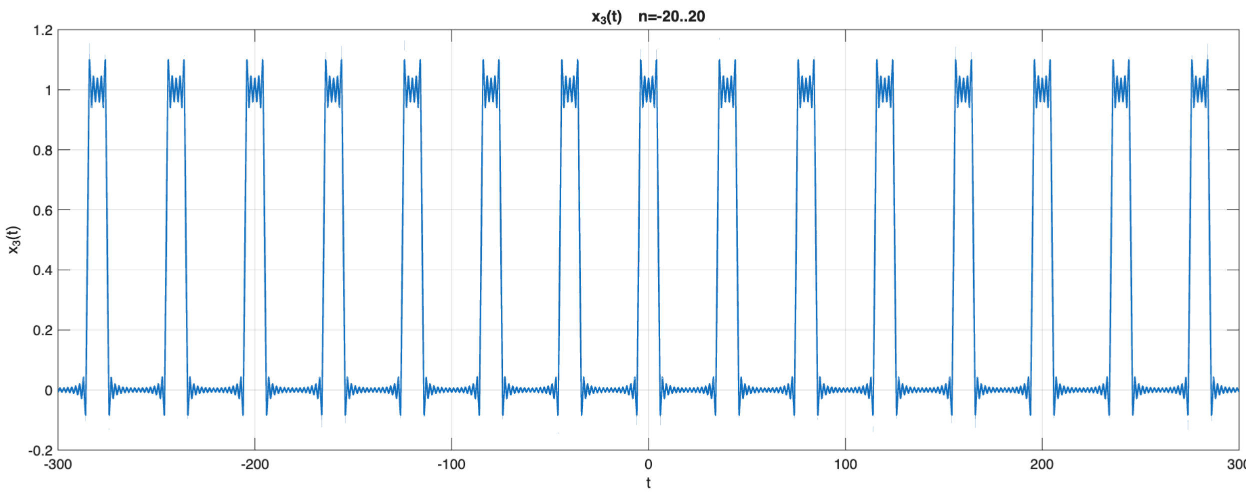
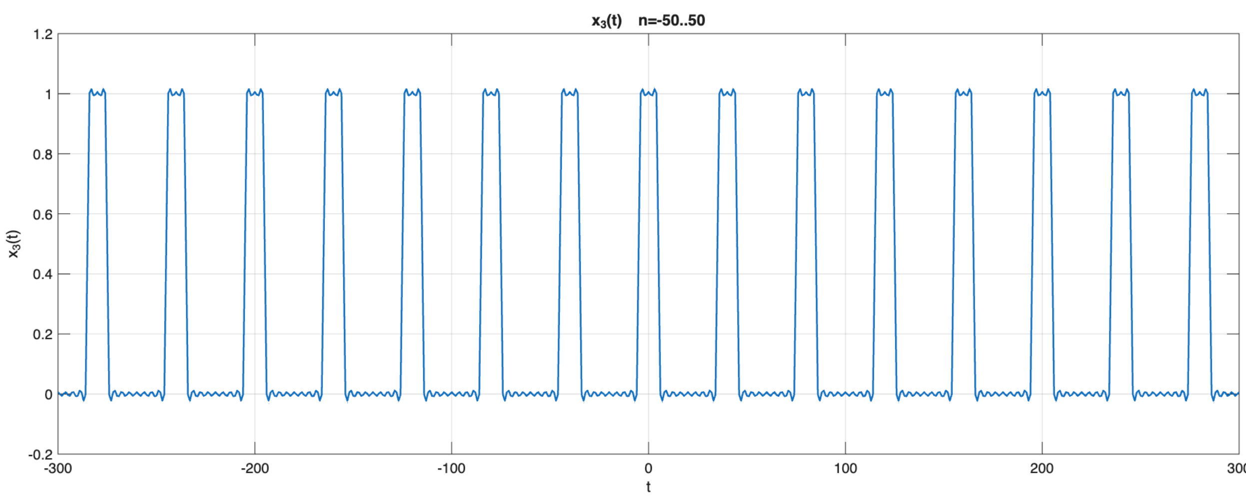


Figure 7: X2(t) Reconstructed


$x_3(t)$ n=-5..5
$x_3(t)$
t
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-300
-200
-100
0
100
200
300


$x_3(t)$ n=-20..20
$x_3(t)$
t
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-300
-200
-100
0
100
200
300


$x_3(t)$ n=-50..50
$x_3(t)$
t
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-300
-200
-100
0
100
200
300

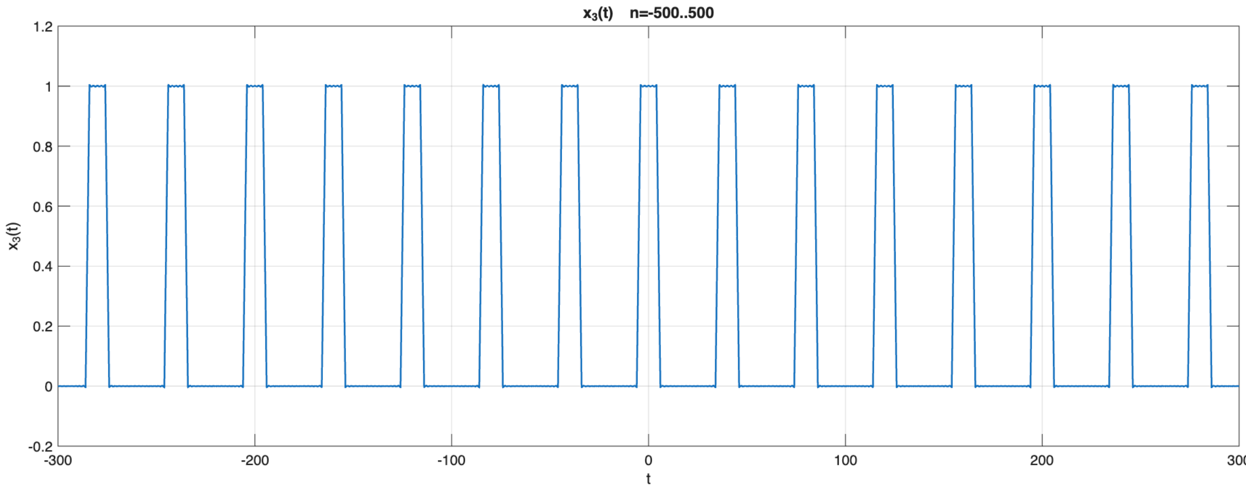


Figure 8: X3(t) Reconstructed

Part B)

B.1

B.1) $W_o = \frac{2\pi}{T_o}$ $\quad x_1(t) = \cos(\frac{3\pi}{10}t) + \frac{1}{2}\cos(\frac{\pi}{10}t)$

Component w are $\frac{3\pi}{10}, \frac{\pi}{10}$ rad/s

$\therefore W_{o,1} = \frac{\pi}{10}$ rad/s (gcd)

$T_{o,1} = \frac{2\pi}{W_{o,1}} = \frac{2\pi}{\pi/10} = 20s$

$x_2(t)$: $T_{o,2} = 20s$ (shown in graph)

$W_{o,2} = \frac{2\pi}{20} = \frac{\pi}{10}$ rad/s

$x_3(t)$: $T_{o,3} = 40s$ (graph)

$W_{o,3} = \frac{2\pi}{40} = \frac{\pi}{20}$ rad/s

B.2

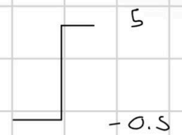
The main difference between the Fourier coefficients of x1(t) and x2(t) is that x1(t) contains only a few discrete frequencies because it is a sum of two cosines, while x2(t) has infinitely many harmonics since it is a rectangular pulse train. The coefficients for x1(t) exist only at certain n values, whereas x2(t) has coefficients that decay gradually following a sinc shaped envelope and include a DC component.

B.3

Despite having the same pulse shape, x2(t) and x3(t) differ in their periods, which have an impact on how far apart their Fourier coefficients are. The fundamental frequency separation between harmonics decreases with increasing period and increases with decreasing period. Consequently, the harmonics of x3(t) are closer together than those of x2(t).

B.4

$x_4(t)$



$T = 20$

Odd signal

$$D_0 = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\,dt$$

$$= \frac{2}{20}\left[\int_0^5 \frac{1}{2}\,dt - \int_5^{15} \frac{1}{2}\,dt + \int_{15}^{20} \frac{1}{2}\,dt\right]$$

$$= \frac{2}{20}\left[\frac{1}{2}t\Big|_0^5 - \frac{1}{2}t\Big|_5^{15} + \frac{1}{2}t\Big|_{15}^{20}\right]$$

$$= \frac{2}{20}(2.5 - 5 + 2.5)$$

$$D_0^{(4)} = 0$$

$x_2(t)$:

$T = 20$

$$D_0 = \frac{1}{T}\int_{-T/2}^{T/2} x_2(t)\,dt$$

$$= \frac{1}{20}\left[\int_{-10}^{-5} 0\,dt + \int_{-5}^{5} 1\,dt + \int_{5}^{10} 0\,dt\right]$$

$$= \frac{1}{20}\left[1t\Big|_{-5}^{5}\right]$$

$$D_0^{(2)} = \frac{1}{20}(10)$$

$$= 0.5$$

$$D_0(x_4) = D_0(x_2) - 0.5 = 0$$

B.5

With an increase in the number of Fourier coefficients, the reconstructed signal approaches the original waveform more closely. Once all of the current harmonics are taken into account, flawless reconstruction for x1(t) happens. The approximation error for x2(t) decreases with increasing coefficients however, the Gibbs phenomenon causes oscillations to persist near the edges. With the addition of more terms, these oscillations become narrower but remain visible.

B.6

A finite number of coefficients is enough for x1(t) because it contains only a few harmonic components. However, x2(t) and x3(t) are discontinuous and therefore require infinitely many coefficients for perfect reconstruction.

B.7

Since a periodic signal generally has an infinite number of Fourier coefficients D, storing all of them is not practical. However, if the signal is finite, such as x1(t), then the corresponding D values can be stored. This approach is not recommended for signals with a large number of significant coefficients, as it would be inefficient and waste memory.

Conclusion

In this lab, MATLAB was used to generate and reconstruct signals using their Fourier series coefficients. The results showed that smooth signals can be represented exactly with a few harmonics, while discontinuous signals require many terms and exhibit Gibbs oscillations. Overall, the lab demonstrated how Fourier series link time-domain periodic signals to their frequency-domain representations.