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

Course Number	ELE 532
Course Title	Signal and Systems 1
Semester/Year	Fall 2023
Instructor	Dr. Javad Alirezaie

Lab/Tutorial Report No.	**3**

Report Title	Fourier Series Analysis using MATLAB

Submission Date	2023-03-21
Due Date	2023-03-21

Name	**Student ID**	**Signature***
Magsud Allahyarov	xxxx84937	

(Note: remove the first 4 digits from your student ID)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

https://www.torontomu.ca/content/dam/senate/policies/pol60.pdf

Table of Contents:

Problem Questions:

Problem A.1:

Given the periodic signal x1(t):

x1(t) = cos3π/10 t + 1/2 cosπ/10 t

derive an expression for the Exponential Fourier Series coefficients Dn.

ELE 532 Lab 3

Wednesday, November 8, 2023 4:38 PM

Maysud Allahyarov

500 984 937

Problem A.1

$$x_1(t) = \cos\frac{3\pi}{10}t + \frac{1}{2}\cos\frac{\pi}{10}t \qquad \cos\omega t = \frac{1}{2}e^{j\omega t} + \frac{1}{2}e^{-j\omega t}$$

$$x_1(t) = \frac{1}{2}\exp\left(j\frac{3\pi}{10}t\right) + \frac{1}{2}\exp\left(-j\frac{3\pi}{10}t\right) + \frac{1}{4}\exp\left(j\frac{\pi}{10}t\right) + \frac{1}{4}\exp\left(-j\frac{\pi}{10}t\right)$$

$$\frac{\frac{3\pi}{10}}{\frac{\pi}{10}} = \frac{3\pi}{10}\cdot\frac{10}{\pi} = 3 \text{ rational}$$

$$\omega_0 = \frac{\pi}{10} \leftarrow \omega_{0_1} = \frac{\pi}{10},\ \omega_{0_2} = \frac{\pi}{10} \qquad T_0 = \frac{2\pi}{\frac{\pi}{10}} = 20$$

$$D_n = \frac{1}{T_0}\int_{T_0} x(t)\exp(-jn\omega_0 t)\,dt$$

$$D_n = \frac{1}{20}\int_{-10}^{10}\left[\frac{1}{2}\exp\left(j\frac{3\pi}{10}t\right) + \frac{1}{2}\exp\left(-j\frac{3\pi}{10}t\right) + \frac{1}{4}\exp\left(j\frac{\pi}{10}t\right) + \frac{1}{4}\exp\left(-j\frac{\pi}{10}t\right)\right]\cdot e^{-j\pi/10\, nt}\,dt$$

$$D_n = \frac{1}{20}\left[\frac{e^{j(3-n)\pi} - e^{-j(3-n)\pi}}{2j(3-n)\frac{\pi}{10}} + \frac{e^{j(3+n)\pi} - e^{-j(3+n)\pi}}{2j(3+n)\frac{\pi}{10}} + \frac{e^{j(1+n)\pi} - e^{-j(1+n)\pi}}{4j(1+n)\frac{\pi}{10}} + \frac{e^{j(1-n)\pi} - e^{-j(1-n)\pi}}{4j(1-n)\frac{\pi}{10}}\right]$$

apply

$$\sin(\omega t) = \frac{1}{2j}e^{j\omega t} - \frac{1}{2j}e^{-j\omega t}$$

$$D_n = \frac{1}{2}\left[\sin(3-n)\pi + \sin(3+n)\pi + \frac{1}{2}\sin(1+n)\pi + \frac{1}{2}\sin(1-n)\pi\right]$$

Problem A.2:

Repeat Problem A.1 for the periodic signals x2(t) and x3(t) shown in Figure 1.

Problem A.2

$x_2(t)$:

$T_0 = 20 \quad \omega_0 = \frac{2\pi}{T_0} = \frac{\pi}{10}$

$$D_n = \frac{1}{20}\int_{-5}^{5}(1)e^{-jn\frac{\pi}{10}t}\,dt = \frac{1}{20}\left[\frac{-1}{jn\frac{\pi}{10}}e^{-jn\frac{\pi}{10}t}\Big|_{-5}^{5}\right] = \frac{1}{20}\left(\frac{-10}{jn\pi}e^{-jn\frac{\pi}{2}} + \frac{10}{jn\pi}e^{jn\frac{\pi}{2}}\right)$$

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

$x_3(t)$:

$T_0 = 40 \quad \omega_0 = \frac{2\pi}{40} = \frac{\pi}{20}$

$$D_n = \frac{1}{40}\int_{-5}^{5}(1)e^{-jn\frac{\pi}{20}t}\,dt = \frac{1}{40}\left[\frac{-20}{jn\pi}e^{jn\frac{\pi}{4}} + \frac{20}{jn\pi}e^{-jn\frac{\pi}{4}}\right)$$

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

Problem A.3:

Now that you have an expression for Dn, write a MATLAB function that generates Dn for a user specified range of values of n.

Problem A.4:

Generate and plot the magnitude and phase spectra of x1(t), x2(t) and x3(t) (using the stem command) from their respective Dn sets for the following index ranges:

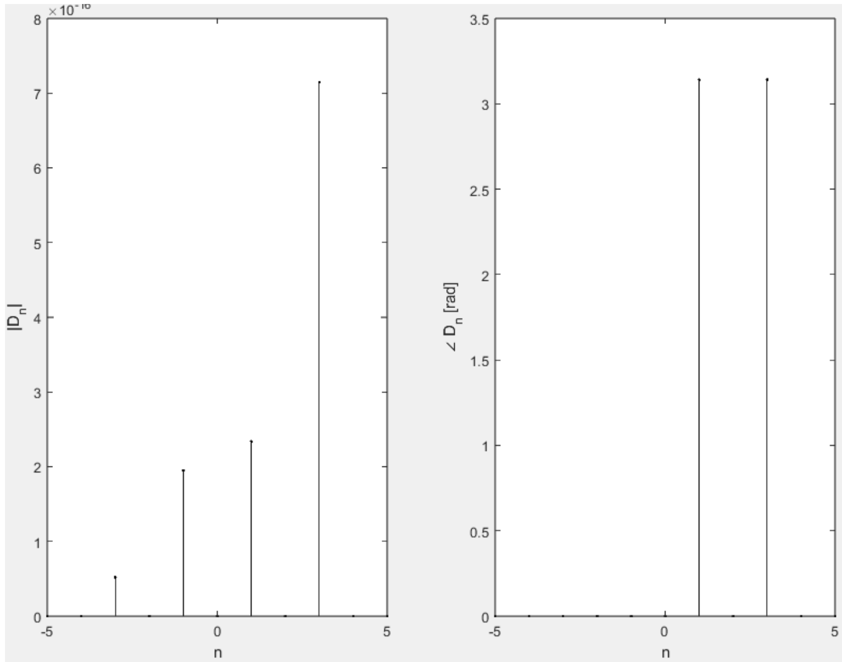
(a) $-5 \leq n \leq 5$;

(b) $-20 \leq n \leq 20$;

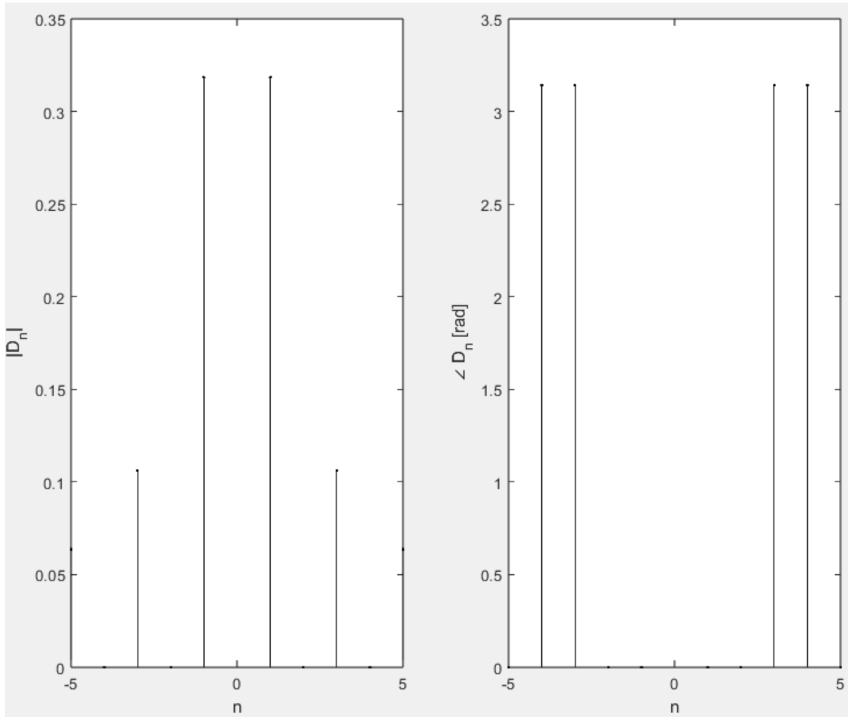
(c) $-50 \leq n \leq 50$;

(d) $-500 \leq n \leq 500$.

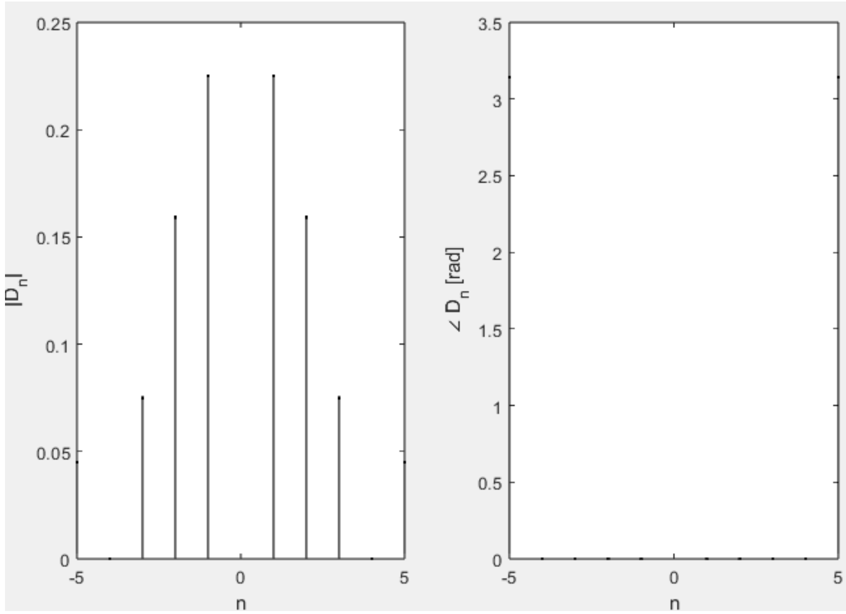
```
%Problem A.4a(x1(t))
clf;
n = (-5:5);
D_n = 1./2.*((1./(pi.*n)).*sin((3-n).*pi)) + (1./pi.*n).*sin((3+n).*pi) +
(1./(2.*n.*pi).*sin((1+n).*pi)) + (1./(2.*n.*pi).*sin((1-n).*pi));
idx = ~(n == 3 | n == -3 | n == 1 | n == -1);
D_n(idx) = 0;
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



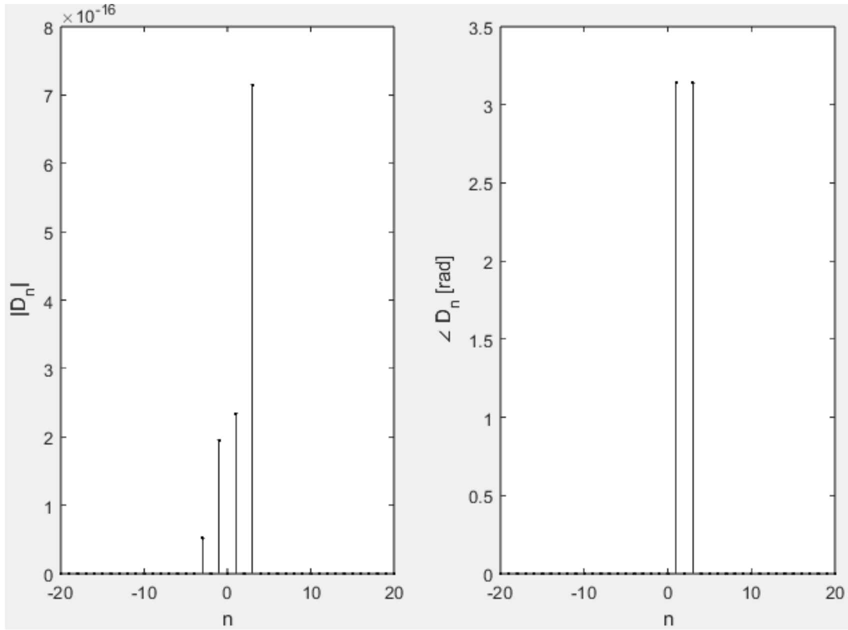

```
%Problem A.4a(x2(t))
clf;
n = (-5:5);
D_n = (1./(n.*pi).*sin((n.*pi)./2));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



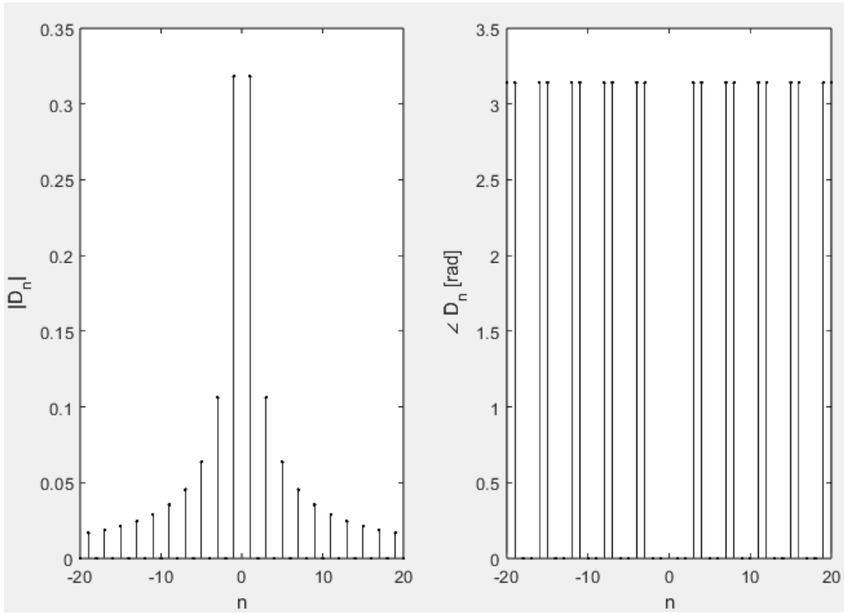

```
%Problem A.4a(x3(t))
clf;
n = (-5:5);
D_n = (1./(n.*pi).*sin((n.*pi)./4));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



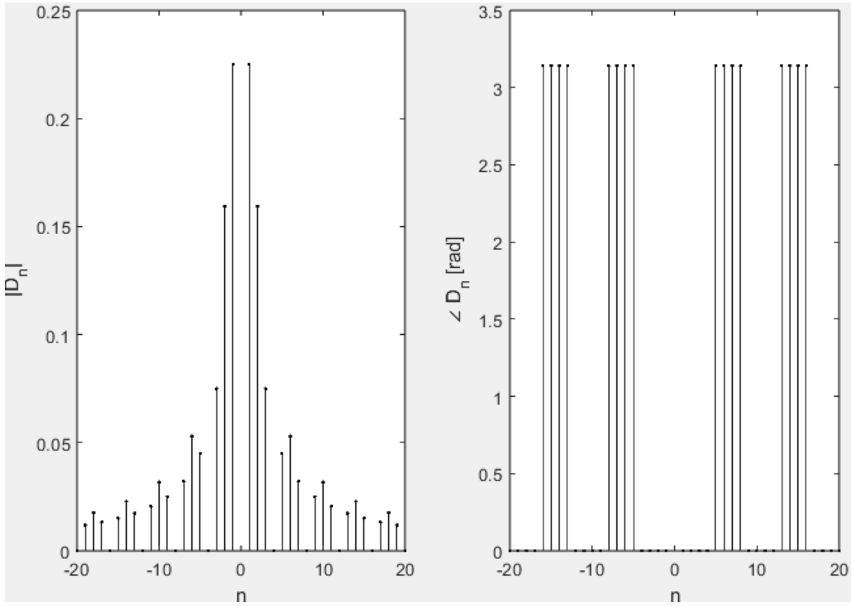

```
%Problem A.4b(x1(t))
clf;
n = (-20:20);
D_n = 1./2.*((1./(pi.*n)).*sin((3-n).*pi)) + (1./pi.*n).*sin((3+n).*pi) +
(1./(2.*n.*pi).*sin((1+n).*pi)) + (1./(2.*n.*pi).*sin((1-n).*pi));
idx = ~(n == 3 | n == -3 | n == 1 | n == -1);
D_n(idx) = 0;
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



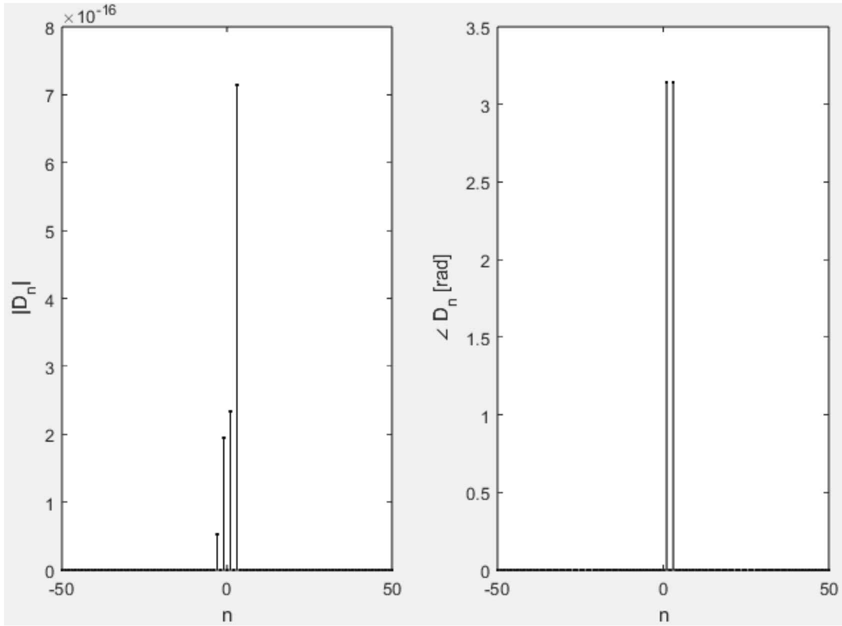

```
%Problem A.4b(x2(t))
clf;
n = (-20:20);
D_n = (1./(n.*pi).*sin((n.*pi)./2));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



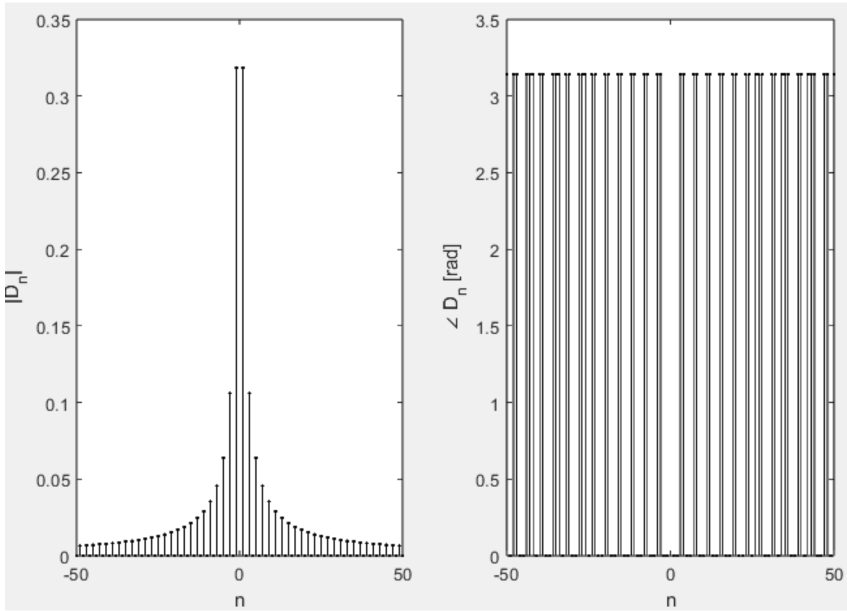

```
%Problem A.4b(x3(t))
clf;
n = (-20:20);
D_n = (1./(n.*pi).*sin((n.*pi)./4));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



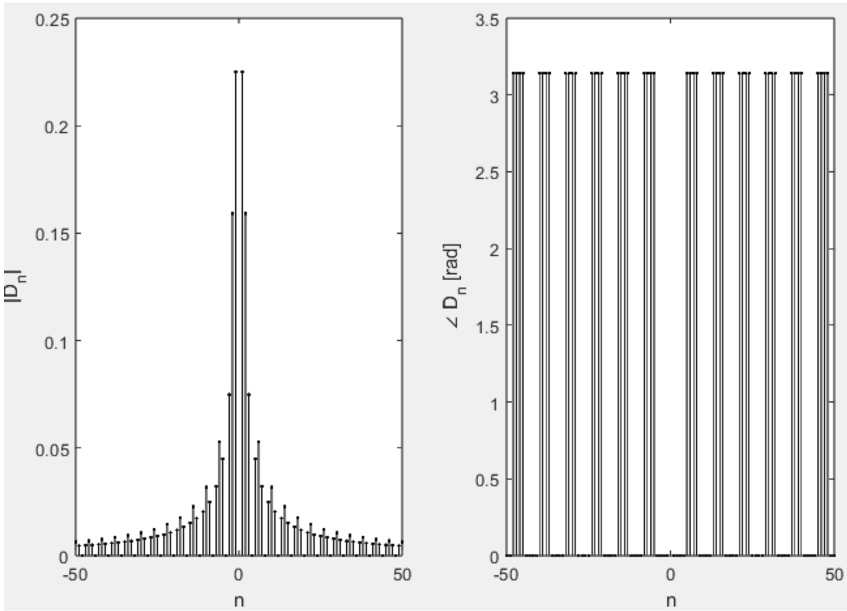

```
%Problem A.4c(x1(t))
clf;
n = (-50:50);
D_n = 1./2.*((1./(pi.*n)).*sin((3-n).*pi)) + (1./pi.*n).*sin((3+n).*pi) +
(1./(2.*n.*pi).*sin((1+n).*pi)) + (1./(2.*n.*pi).*sin((1-n).*pi));
idx = ~(n == 3 | n == -3 | n == 1 | n == -1);
D_n(idx) = 0;
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



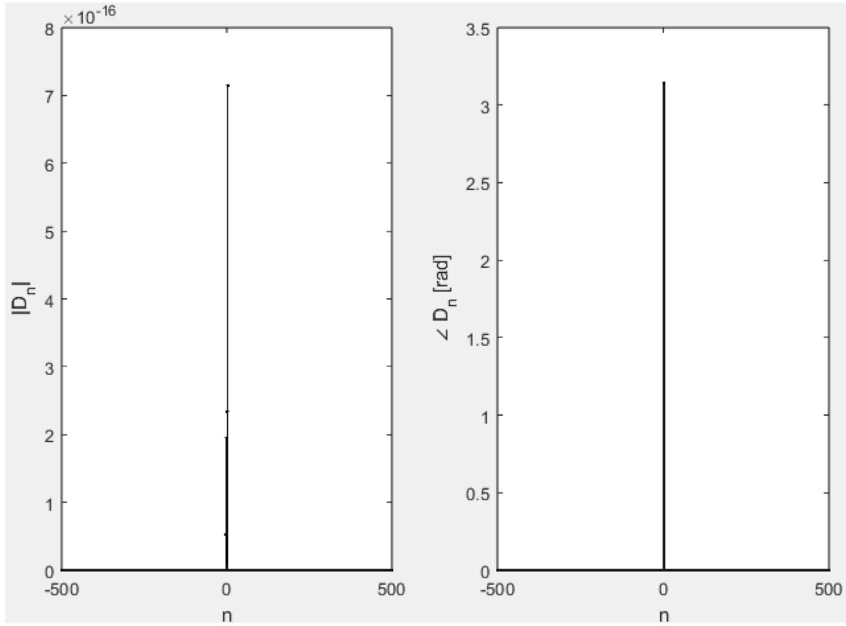

```
%Problem A.4c(x2(t))
clf;
n = (-50:50);
D_n = (1./(n.*pi).*sin((n.*pi)./2));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



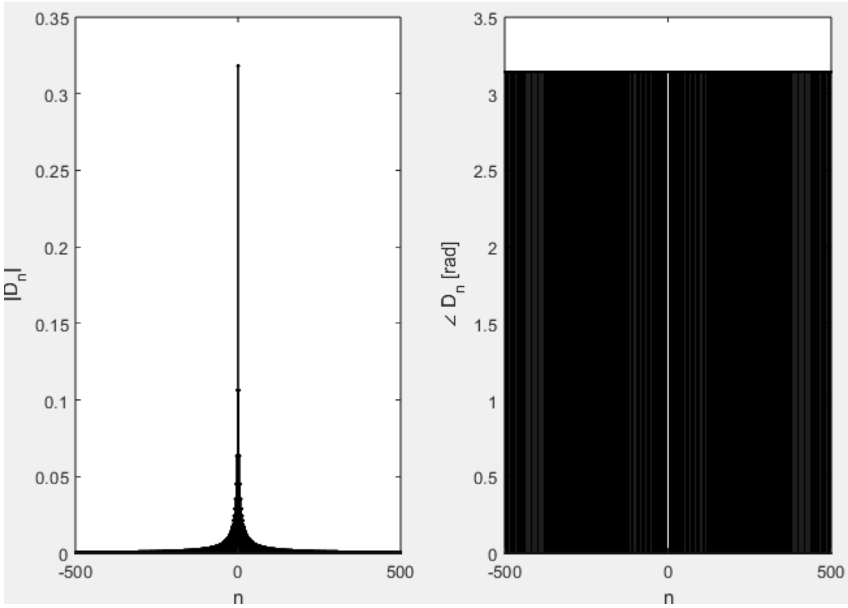

```
%Problem A.4c(x3(t))
clf;
n = (-50:50);
D_n = (1./(n.*pi).*sin((n.*pi)./4));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



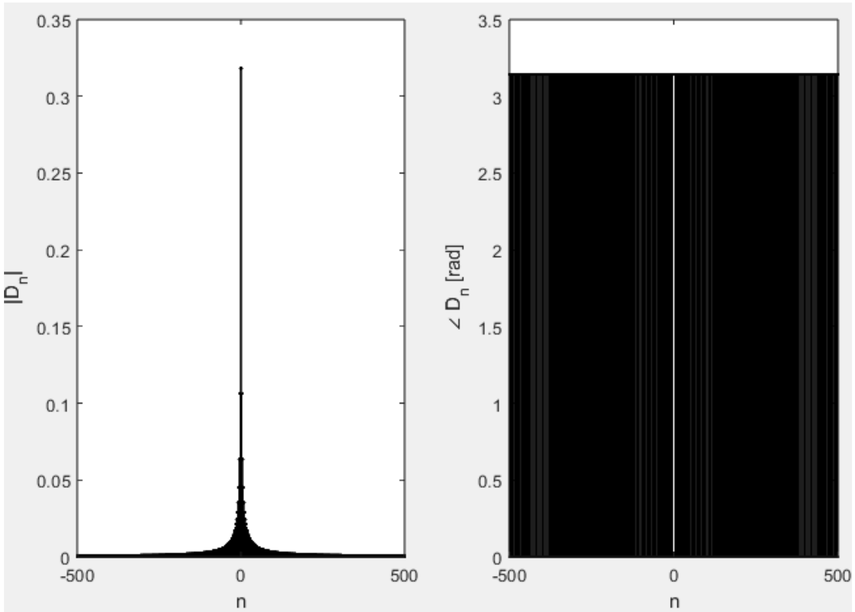

```
%Problem A.4d(x1(t))
clf;
n = (-500:500);
D_n = 1./2.*((1./(pi.*n)).*sin((3-n).*pi)) + (1./pi.*n).*sin((3+n).*pi) +
(1./(2.*n.*pi).*sin((1+n).*pi)) + (1./(2.*n.*pi).*sin((1-n).*pi));
idx = ~(n == 3 | n == -3 | n == 1 | n == -1);
D_n(idx) = 0;
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```




```
%Problem A.4d(x2(t))
clf;
n = (-500:500);
D_n = (1./(n.*pi).*sin((n.*pi)./2));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```




```
%Problem A.4d(x3(t))
clf;
n = (-500:500);
D_n = (1./(n.*pi).*sin((n.*pi)./4));
subplot(1,2,1); stem(n,abs(D_n),'.k');
xlabel('n'); ylabel('|D_n|');
subplot(1,2,2); stem(n,angle(D_n),'.k');
xlabel('n'); ylabel('\angle D_n [rad]');
```



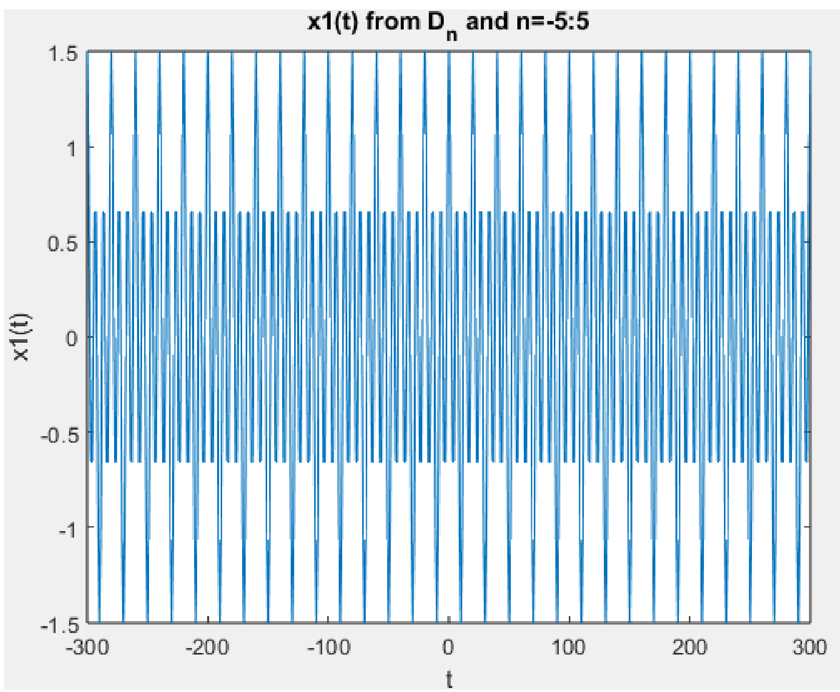
Problem A.5/A.6:

Write a MATLAB function that takes a MATLAB generated Dn set and reconstructs the original time-domain signal from which the Fourier coefficients had been derived. For example, given the set of truncated Fourier coefficients {Dn, n = 0, ±1, . . . , ±20}, your code should reconstruct the time-domain signal from this set using Equation (1). Note: Use the time variable t defined with the MATLAB command t=[-300:1:300].

Reconstruct the time-domain signals x1(t), x2(t) and x3(t) with the Fourier coefficient sets you generated in Problem A.4. Plot each reconstructed signal.

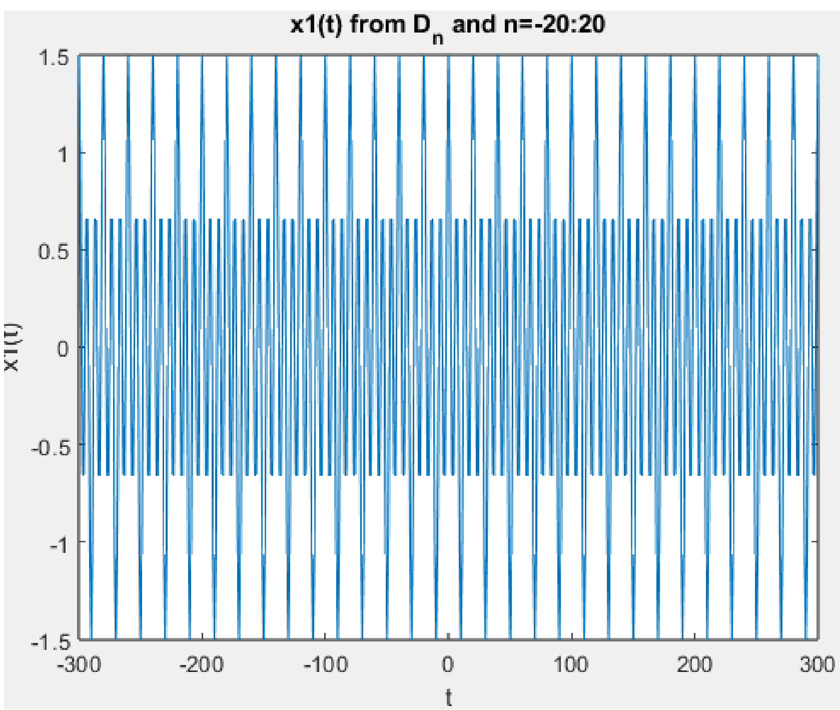
```
%Problem A.5/6a(x1(t))
clf
t = -300:1:300;
x = 0;
for n = -5:5
  D_n = 0;
```

```
  if(n==3 || n==-3)
     D_n=(1/2);
  end
  if(n==1||n==-1)
     D_n=(1/4);
  end
  x=x+D_n.*(exp(sqrt(-1)*n*(pi/10)*t));
end
plot(t,x);
xlabel('t');
ylabel('x1(t)');
title('x1(t) from D_n and n=-5:5')
```



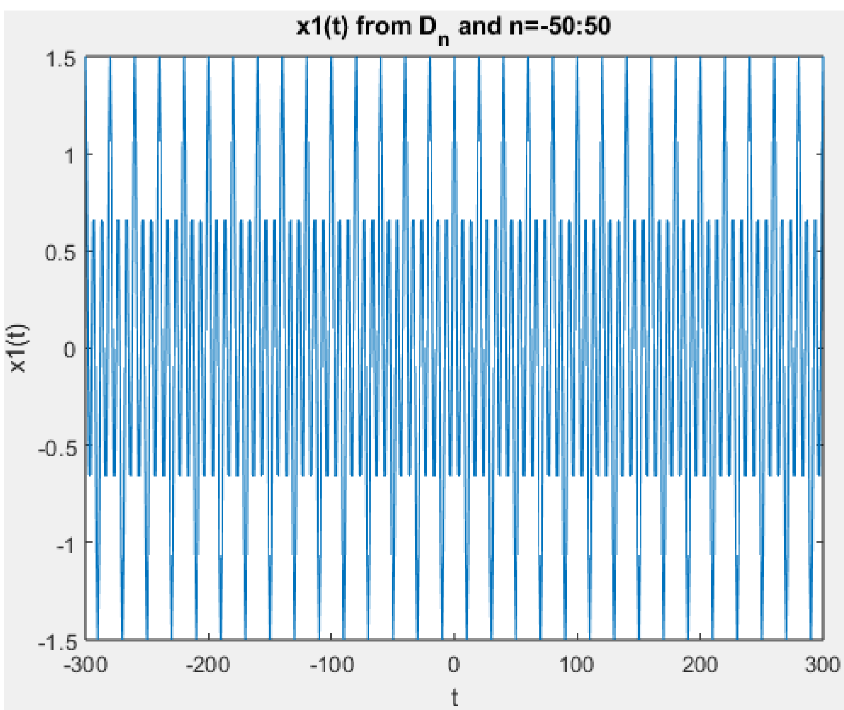

```
%Problem A.5/6b(x1(t))
clf
t = -300:1:300;
x = 0;
```

```
for n = -20:20
  D_n = 0;
  if(n==3 || n==-3)
     D_n=(1/2);
  end
  if(n==1||n==-1)
     D_n=(1/4);
  end
  x=x+D_n.*(exp(sqrt(-1)*n*(pi/10)*t));
end
plot(t,x);
xlabel('t');
ylabel('x1(t)');
title('x1(t) from D_n and n=-20:20')
```

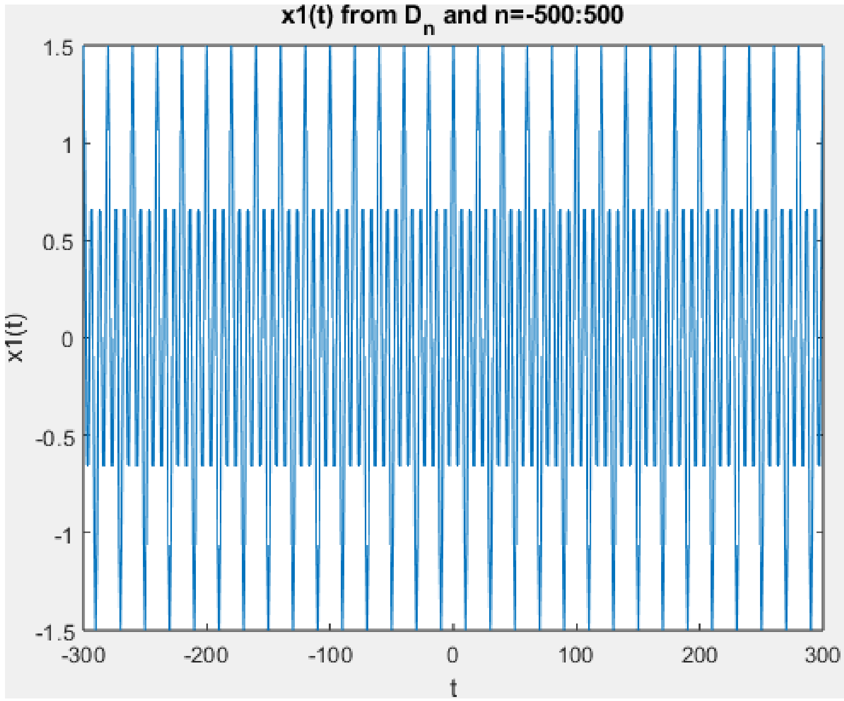



```
%Problem A.5/6c(x1(t))
clf
```

```
t = -300:1:300;
x = 0;
for n = -50:50
  D_n = 0;
  if(n==3 || n==-3)
     D_n=(1/2);
  end
  if(n==1||n==-1)
     D_n=(1/4);
  end
  x=x+D_n.*(exp(sqrt(-1)*n*(pi/10)*t));
end
plot(t,x);
xlabel('t');
ylabel('x1(t)');
title('x1(t) from D_n and n=-50:50')
```

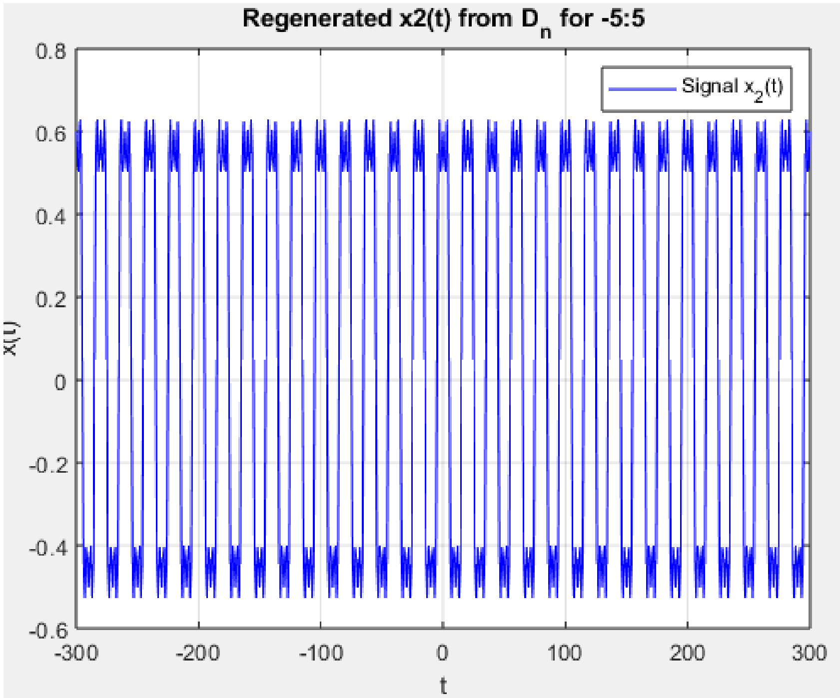


```
%Problem A.5/6d(x1(t))
clf
t = -300:1:300;
x = 0;
for n = -500:500
  D_n = 0;
  if(n==3 || n==-3)
     D_n=(1/2);
  end
  if(n==1||n==-1)
     D_n=(1/4);
  end
  x=x+D_n.*(exp(sqrt(-1)*n*(pi/10)*t));
end
plot(t,x);
xlabel('t');
ylabel('x1(t)');
title('x1(t) from D_n and n=-500:500')
```



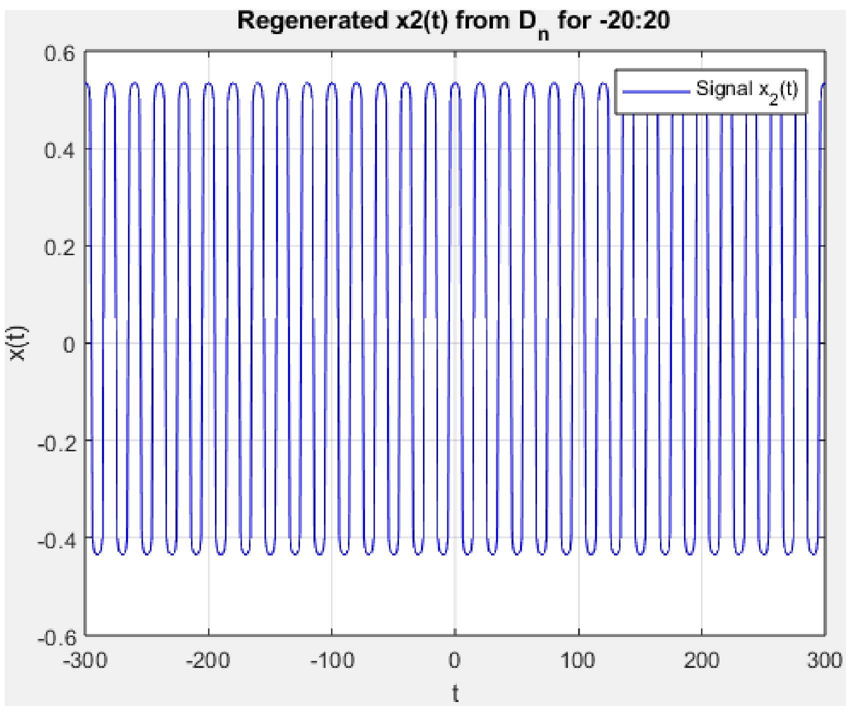

```
%Problem A.5/6a(x2(t))
D_n=[-5:5]; nleftlim = -5; nrightlim = 5; x = 5+1;
for n = [nleftlim:nrightlim];
  if n == 0,
     D_n(x) = 0.05;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.5)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/10; t = -300:300;
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
```

```
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x2(t) from D_n for -5:5');
legend('Signal x_2(t)'); grid;
```



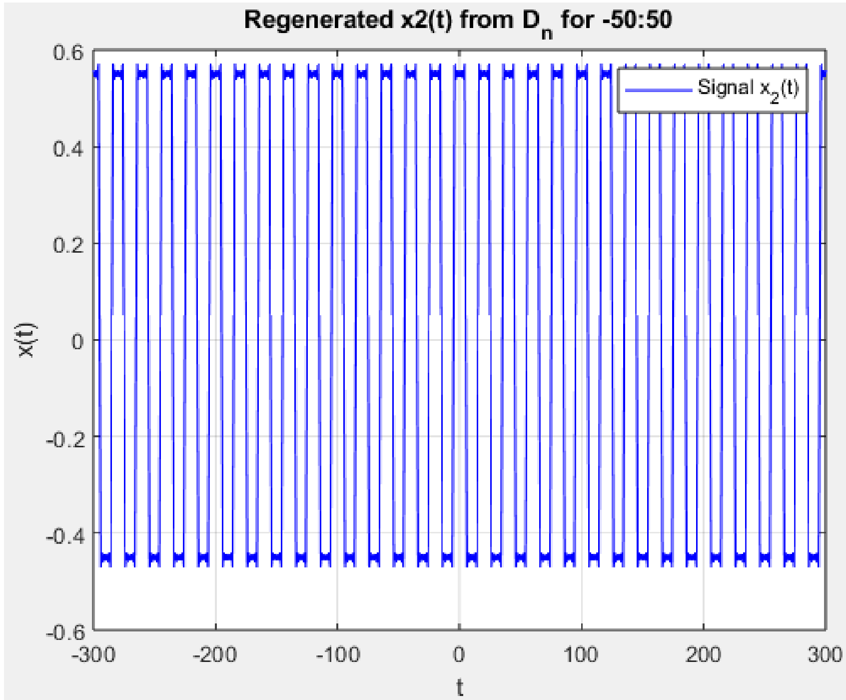

```
%Problem A.5/6b(x2(t))
D_n=[-20:20]; nleftlim = -20; nrightlim = 20; x = 20+1;
for n = [nleftlim:nrightlim];
  if n == 0,
     D_n(x) = 0.05;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.5)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/10; t = -300:300;
```

```
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x2(t) from D_n for -20:20');
legend('Signal x_2(t)'); grid;
```



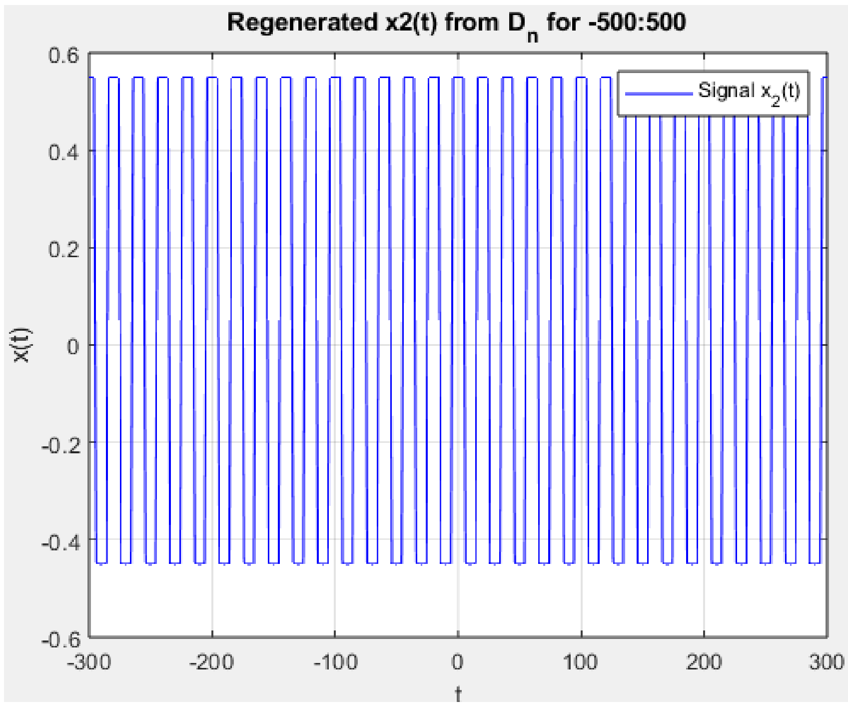

```
%Problem A.5/6c(x2(t))
D_n=[-50:50]; nleftlim = -50; nrightlim = 50; x = 50+1;
for n = [nleftlim:nrightlim];
```

```
  if n == 0,
     D_n(x) = 0.05;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.5)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/10; t = -300:300;
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x2(t) from D_n for -50:50');
legend('Signal x_2(t)'); grid;
```



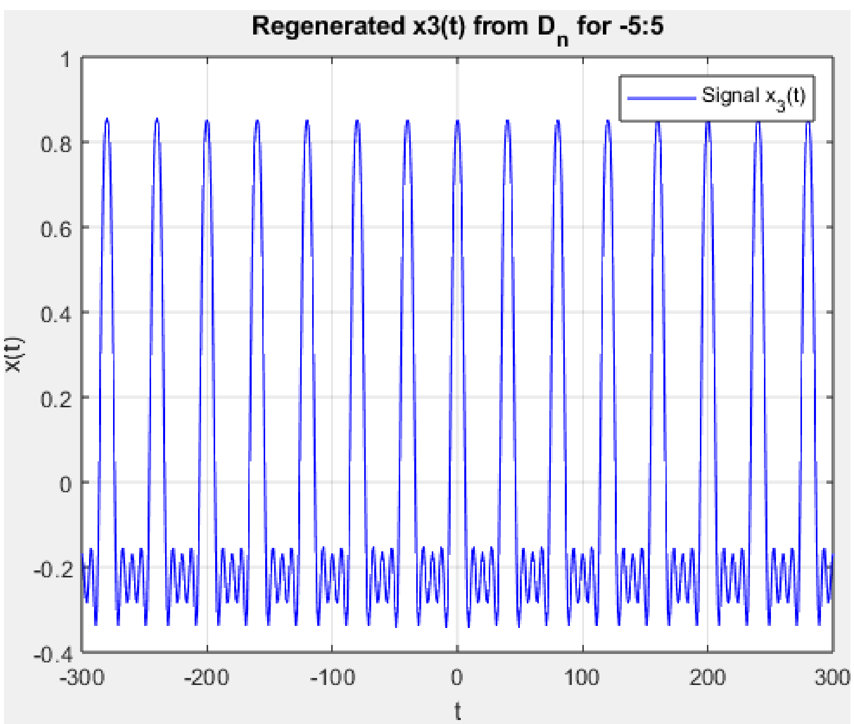

```
%Problem A.5/6d(x2(t))
D_n=[-500:500]; nleftlim = -500; nrightlim = 500; x = 500+1;
for n = [nleftlim:nrightlim];
  if n == 0,
      D_n(x) = 0.05;
  else
      D_n(n-nleftlim+1) = (sin(n.*pi*0.5)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/10; t = -300:300;
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
      x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
```

```
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x2(t) from D_n for -500:500');
legend('Signal x_2(t)'); grid;
```



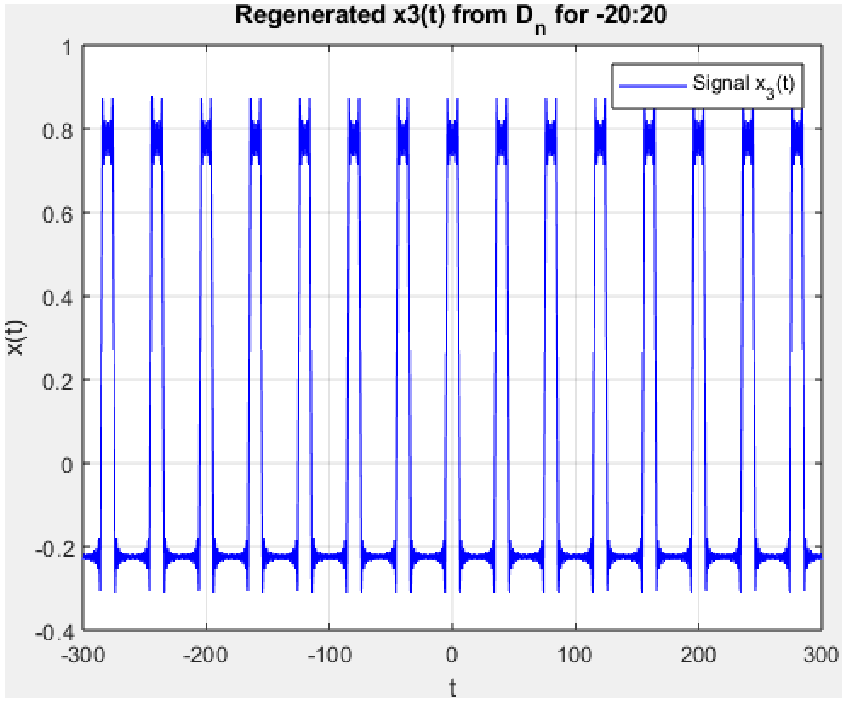

```
%Problem A.5/6a(x3(t))
D_n=[-5:5]; nleftlim = -5; nrightlim = 5; x = 5+1;
for n = [nleftlim:nrightlim];
  if n == 0,
     D_n(x) = 0.025;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.25)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/20; t = -300:300;
```

```
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
      x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x3(t) from D_n for -5:5');
legend('Signal x_3(t)'); grid;
```



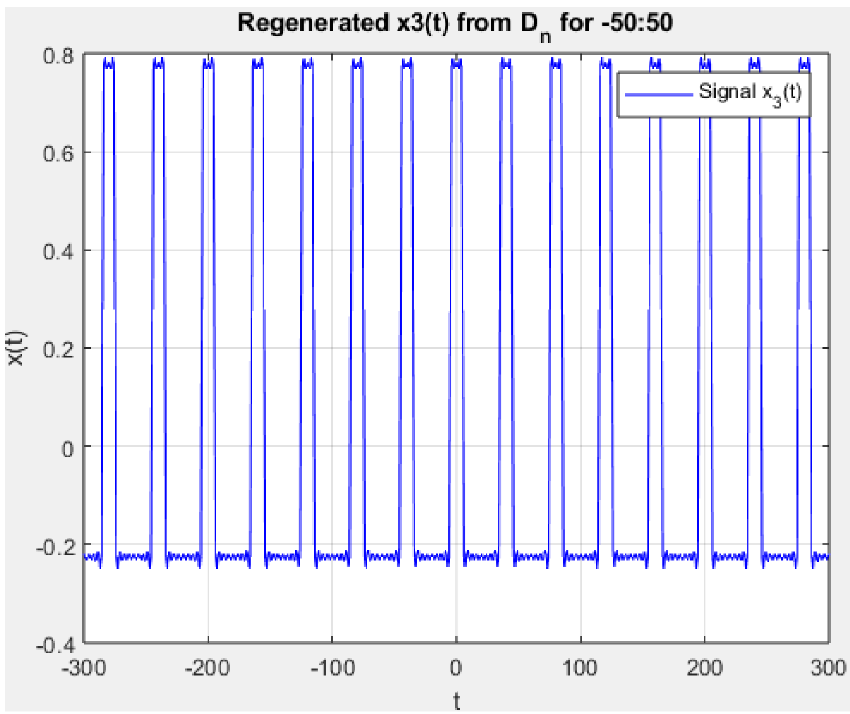

```
%Problem A.5/6b(x3(t))
D_n=[-20:20]; nleftlim = -20; nrightlim = 20; x = 20+1;
for n = [nleftlim:nrightlim];
```

```
  if n == 0,
     D_n(x) = 0.025;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.25)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/20; t = -300:300;
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x3(t) from D_n for -20:20');
legend('Signal x_3(t)'); grid;
```



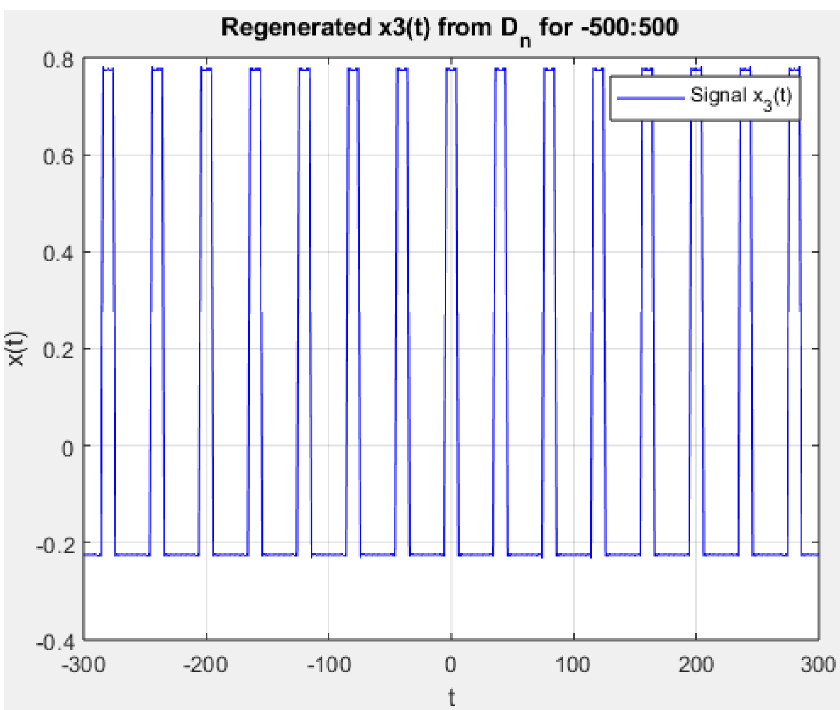

```
%Problem A.5/6c(x3(t))
D_n=[-50:50]; nleftlim = -50; nrightlim = 50; x = 50+1;
for n = [nleftlim:nrightlim];
  if n == 0,
     D_n(x) = 0.025;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.25)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/20; t = -300:300;
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
```

```
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x3(t) from D_n for -50:50');
legend('Signal x_3(t)'); grid;
```




```
%Problem A.5/6d(x3(t))
D_n=[-500:500]; nleftlim = -500; nrightlim = 500; x = 500+1;
for n = [nleftlim:nrightlim];
  if n == 0,
     D_n(x) = 0.025;
  else
     D_n(n-nleftlim+1) = (sin(n.*pi*0.25)./(n.*pi));
  end
end
n = [nleftlim:nrightlim]; W0 = pi/20; t = -300:300;
```

```
s = 300+1;
b=length(t);
x = zeros(1,b);
for t=-300:300
  for n=nleftlim:nrightlim
     x(t+s) = x(t+s) + real(D_n(n-nleftlim+1).*exp(n.*1i*W0*t));
  end
end
t=-300:300;
plot(t,real(x),'b');
ylabel('x(t)'); xlabel('t');
title('Regenerated x3(t) from D_n for -500:500');
legend('Signal x_3(t)'); grid;
```



Discussion

Problem B.1:

Determine the fundamental frequencies of x1(t), x2(t) and x3(t)

$$w_0 = \frac{GCF\ of\ numerator}{LCM\ of\ denominator}$$

or

$$w_0 = \frac{2\pi}{T_0}$$

$x_1(t)$:

$$w_{01} = \frac{3\pi}{10}, w_{02} = \frac{\pi}{10}$$

so

$$w_0 = \frac{\pi}{10}$$

$x_2(t)$:

$$T_0 = 20$$

$$w_0 = \frac{\pi}{10}$$

$x_3(t)$:

$$T_0 = 40$$

$$w_0 = \frac{\pi}{20}$$

Problem B.2:

What is the main difference between the Fourier coefficients of x1(t) and x2(t)?

The main differences between the fourier coefficients of x1(t) and x2(t) is that x1(t) has four distinct fourier series coefficients while x2(t) has infinite fourier coefficients for Dn.

Problem B.3:

Signals x2(t) and x3(t) have the same rectangular pulse shape but different periods. How are these characteristics reflected in their respective Fourier coefficients?

For its Fourier coefficients, signal x3(t) has a lower fundamental frequency value than signal x2(t).

Problem B.4:

The Fourier coefficient D0 represents the DC value of the signal. Let x4(t) be the periodic waveform shown in Figure 2. Derive D0 of x4(t) from D0 of x2(t).

Derivation of signal x4(t) from x2(t) yields D0=0.5.

Problem B.5:

Using the results of Problem A.6, explain how the reconstructed signal changes as you increase the number of Fourier coefficients used in the reconstruction. Discuss for both x1(t) and x2(t).

Increasing the Fourier coefficients won't do anything because x1(t) has a finite number of Dn values. But for x2(t) and x3(t) increasing values of Dn will provide greater accuracy.

Problem B.6:

How many Fourier coefficients do you need to perfectly reconstruct the periodic waveforms discussed in this lab experiment?

Since x1(t) has a finite number of DN values only the four Fourier series coefficients are needed to perfectly reconstruct in this case. For x2(t) and x3(t) on the other hand we would require an infinite number for perfect reconstruction.

Problem B.7:

Let x(t) be an arbitrary periodic signal. Instead of storing x(t) on a computer, we consider storing the corresponding Fourier coefficients. When we need to access x(t), we read the Fourier coefficients stored on the computer hard drive and reconstruct the signal. Is this a viable scenario? Explain your answer.

A periodic signal is not viable since it has an endless number of Dn values. In case it is finite like with x1(t) then it is possible to store the values of Dn. However, this is not recommended for signals which have a large amount of finite Dn values as it would just waste space.