Name: Do Trung Hau

Section:

Laboratory Exercise 1

DISCRETE-TIME SIGNALS: TIME-DOMAIN REPRESENTATION

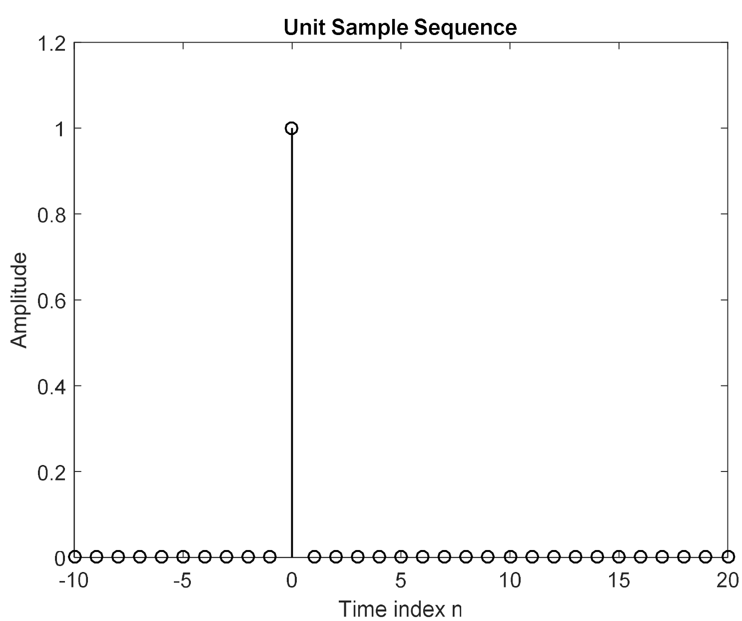
1.1 GENERATION OF SEQUENCES

Project 1.1 Unit sample and unit step sequences

A copy of Program P1_1 is given below.

```
% Generation of a Unit Sample Sequence
clc; clear all; close all;
% Generate a vector from -10 to 20
n = -10:20;
% Generate the unit sample sequence
u = [zeros(1,10) 1 zeros(1,20)];
% Plot the unit sample sequence
stem(n,u,'k','LineWidth', 1);
xlabel('Time index n');ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-10 20 0 1.2]);
```

Q1.1 The unit sample sequence u[n] generated by running Program P1_1 is shown below:



Q1.2 The purpose of `clf` command is used to ***clear current figure window.***

The purpose of `axis` command is used to ***set axis limits and appearance.***

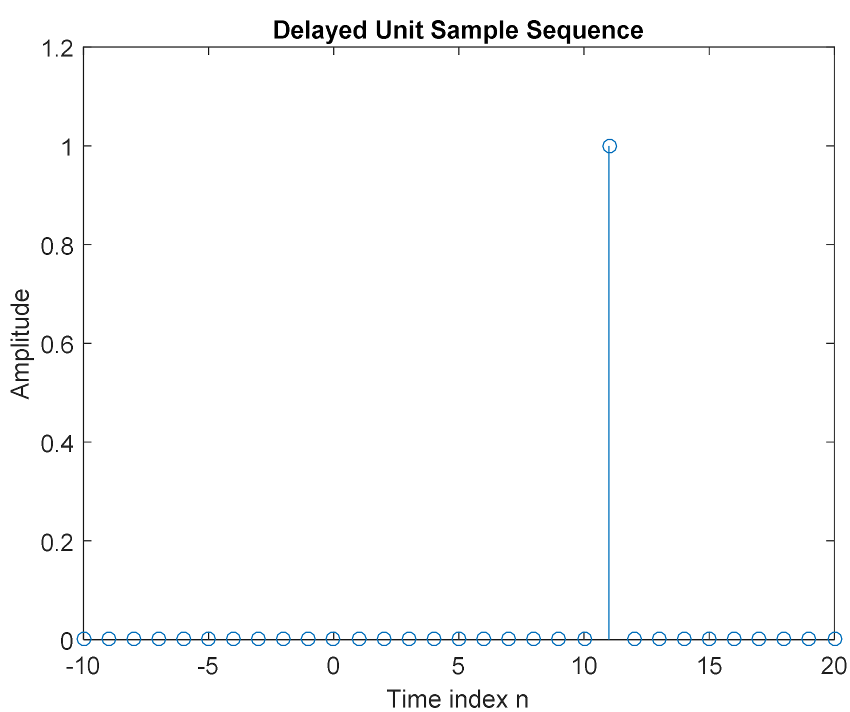
The purpose of `title` command is used to ***add title to axes or legend.***

The purpose of `xlabel` command is used to ***label x-axis.***

The purpose of `ylabel` command is used to ***label y-axis.***

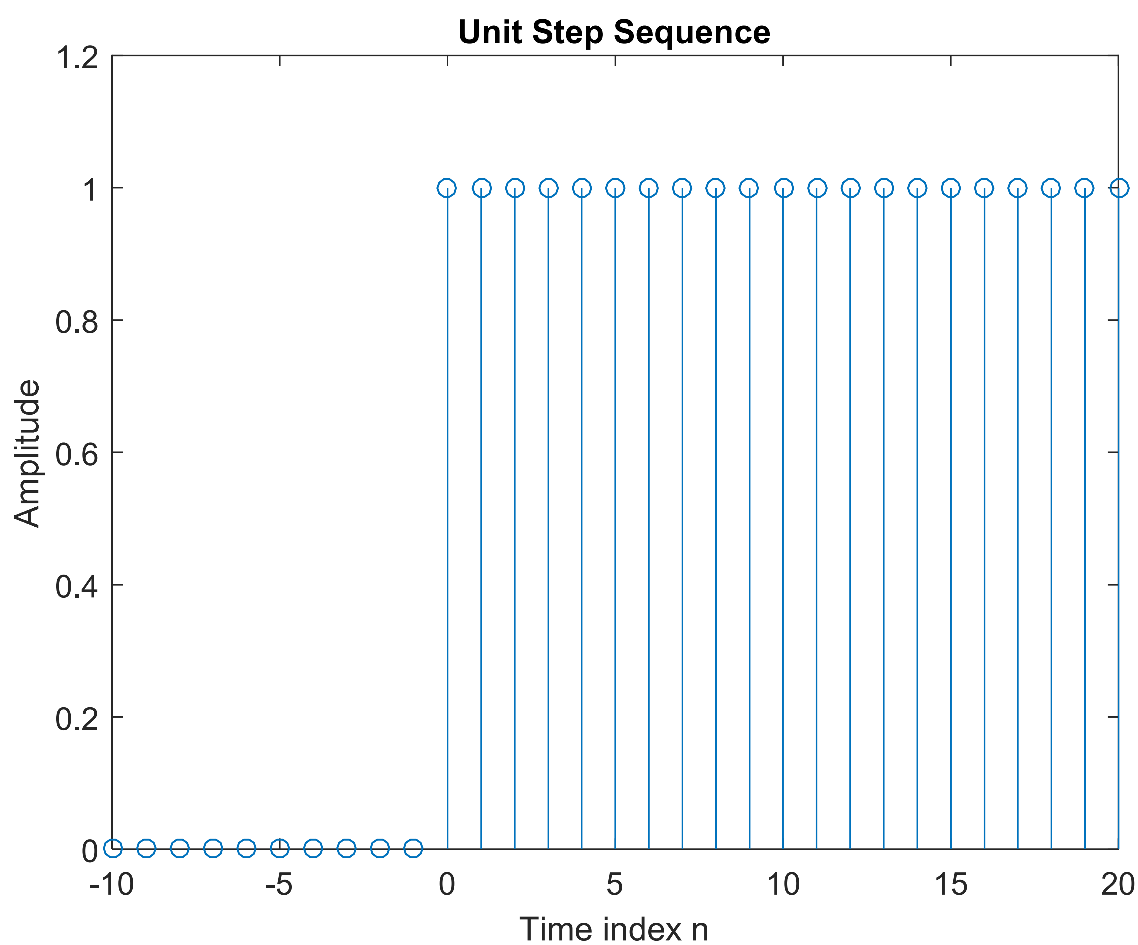
Q1.3 The modified Program P1_1 to generate a delayed unit sample sequence ud[n] with a ***delay*** of 11 samples is given below along with the sequence generated by running this program.

```
% Generation of a Unit Sample Sequence
clc; clear all; close all;
% Generate a vector from -10 to 20
n = -10:20;
% Generate the delayed unit sample sequence
u = [zeros(1,21) 1 zeros(1,9)];
% Plot the unit sample sequence
stem(n,u);
xlabel('Time index n');ylabel('Amplitude');
title('Delayed Unit Sample Sequence');
axis([-10 20 0 1.2]);
```



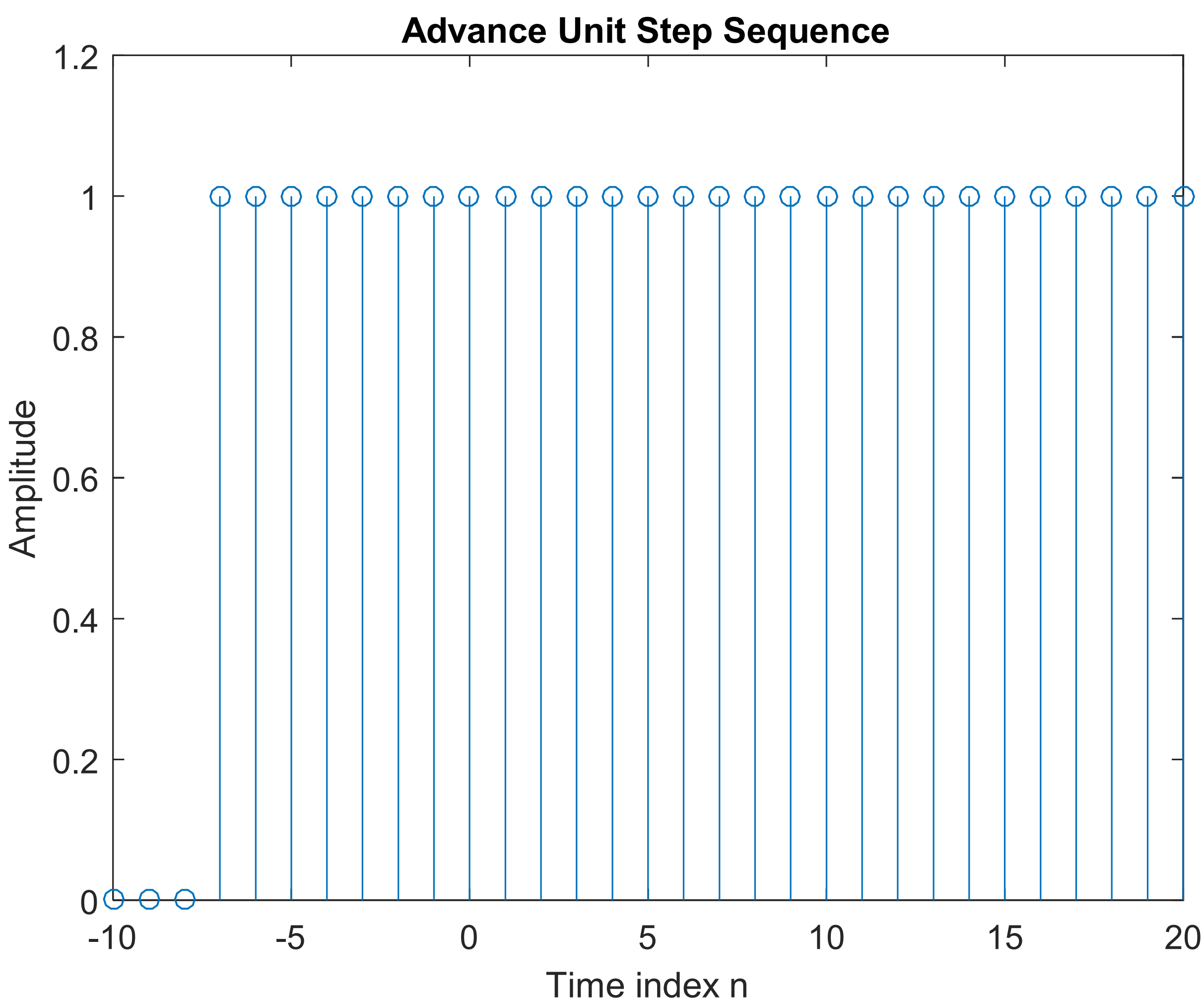
Q1.4 The modified Program P1_1 to generate ***a unit step sequence*** s[n] is given below along with the sequence generated by running this program.

```
% Generation of a Unit Sample Sequence
clc; clear all; close all;
% Generate a vector from -10 to 20
n = -10:20;
% Generate the unit sample sequence
u = [zeros(1,10) ones(1,21)];
% Plot the unit sample sequence
stem(n,u);
xlabel('Time index n');ylabel('Amplitude');
title('Unit Step Sequence');
axis([-10 20 0 1.2]);
```



Q1.5 The modified Program P1_1 to generate a unit step sequence sd[n] with ***an advance of 7 samples*** is given below along with the sequence generated by running this program.

```
clc; clear all; close all;
n = -10:20;
% Generate the unit sample sequence
u = [zeros(1,3) ones(1,28)];
% Plot the unit sample sequence
stem(n,u);
xlabel('Time index n');ylabel('Amplitude');
title('Advance Unit Step Sequence');
axis([-10 20 0 1.2]);
```



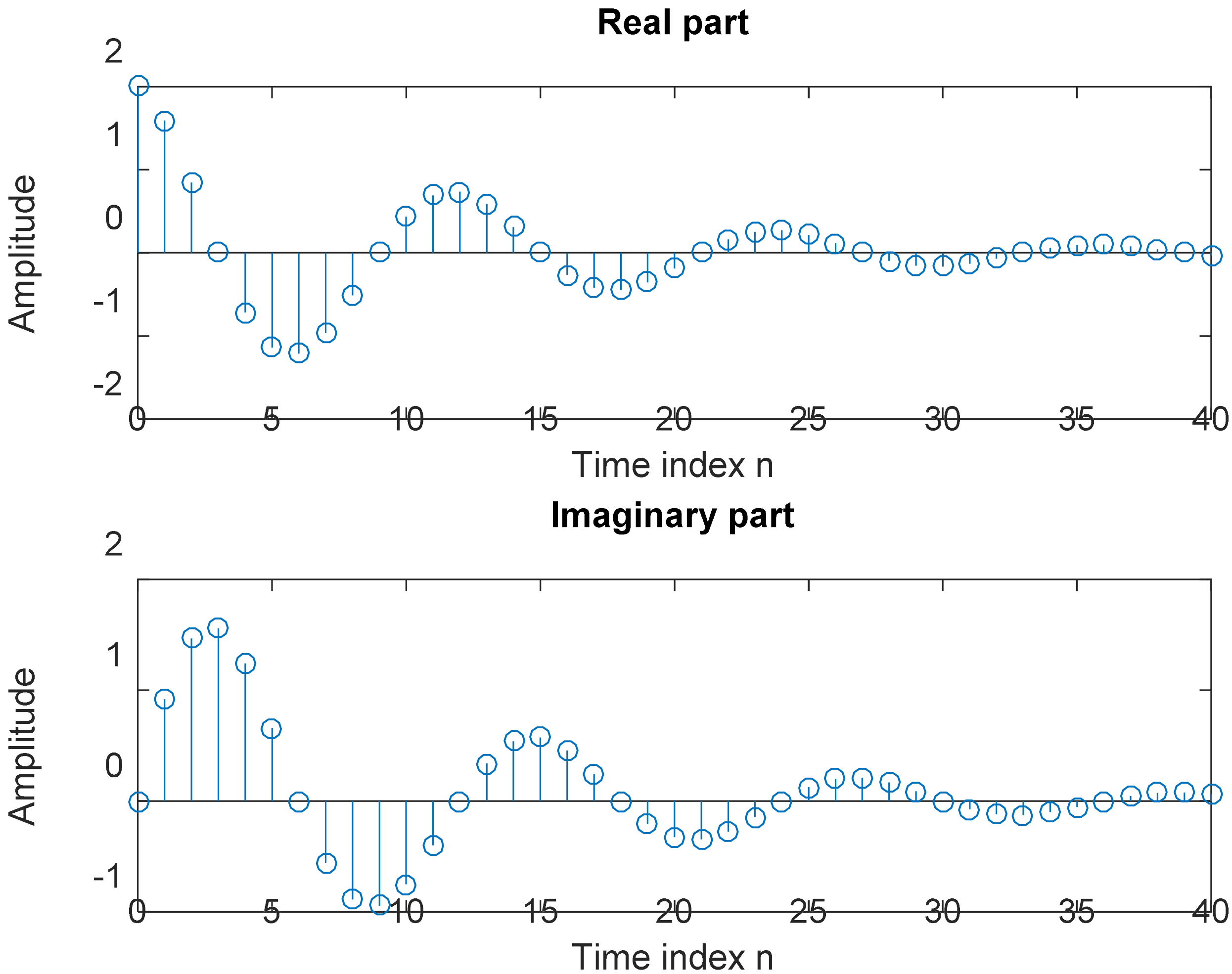
Project 1.2 Exponential signals

A copy of Programs P1_2 and P1_3 are given below.

```
% Generation of a complex exponential
sequence
clc; clear all; close all;
c = -(1/12)+(pi/6)*i;
K = 2;
n = 0:40;
x = K*exp(c*n);
subplot(2,1,1);
stem(n,real(x));
xlabel('Time index n');ylabel('Amplitude');
title('Real part');
subplot(2,1,2);
stem(n,imag(x));
xlabel('Time index n');ylabel('Amplitude');
title('Imaginary part');
```

```
% Generation of a real exponential sequence
clc; clear all; close all;
n = 0:35; a = 1.2; K = 0.2;
x = K*a.^n;
stem(n,x);
xlabel('Time index n');ylabel('Amplitude');
```

Q1.6 The complex-valued exponential sequence generated by running Program P1_2 is shown below:



Q1.7 The parameter controlling the rate of growth or decay of this sequence is ***the real part of parameter "C".***

The parameter controlling the amplitude of this sequence is ***parameter "K".***

Q1.8 The result of changing the parameter c to (1/12)+(pi/6)*i is:

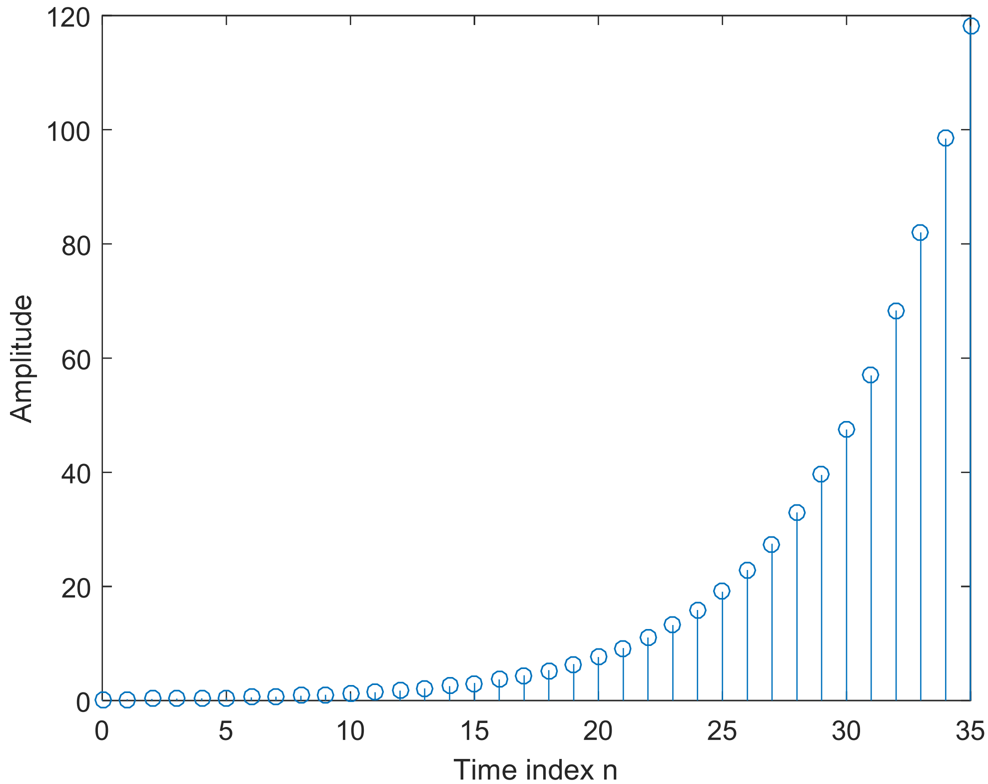
Since the exponential function now has $e^{\frac{1}{12}} \approx 1.087 > 1$ ***, it means that the exponential is increasing, so the signal now has an extended envelope of n (divergence). In contrast to the case*** $c = -\frac{1}{12} + \frac{\pi}{6}j$ ***with*** $e^{-\frac{1}{12}} \approx 0.92 < 1$***, we have the envelope narrowing to n (convergence).***

Q1.9 The purpose of the operator `real` is used ***to get the real part of a vector.***

The purpose of the operator `imag` is ***used to get the imaginary of a vector.***

Q1.10 The purpose of the command `subplot` is ***used to create axes in tiled positions.***

Q1.11 The real-valued exponential sequence generated by running Program P1_3 is shown below:



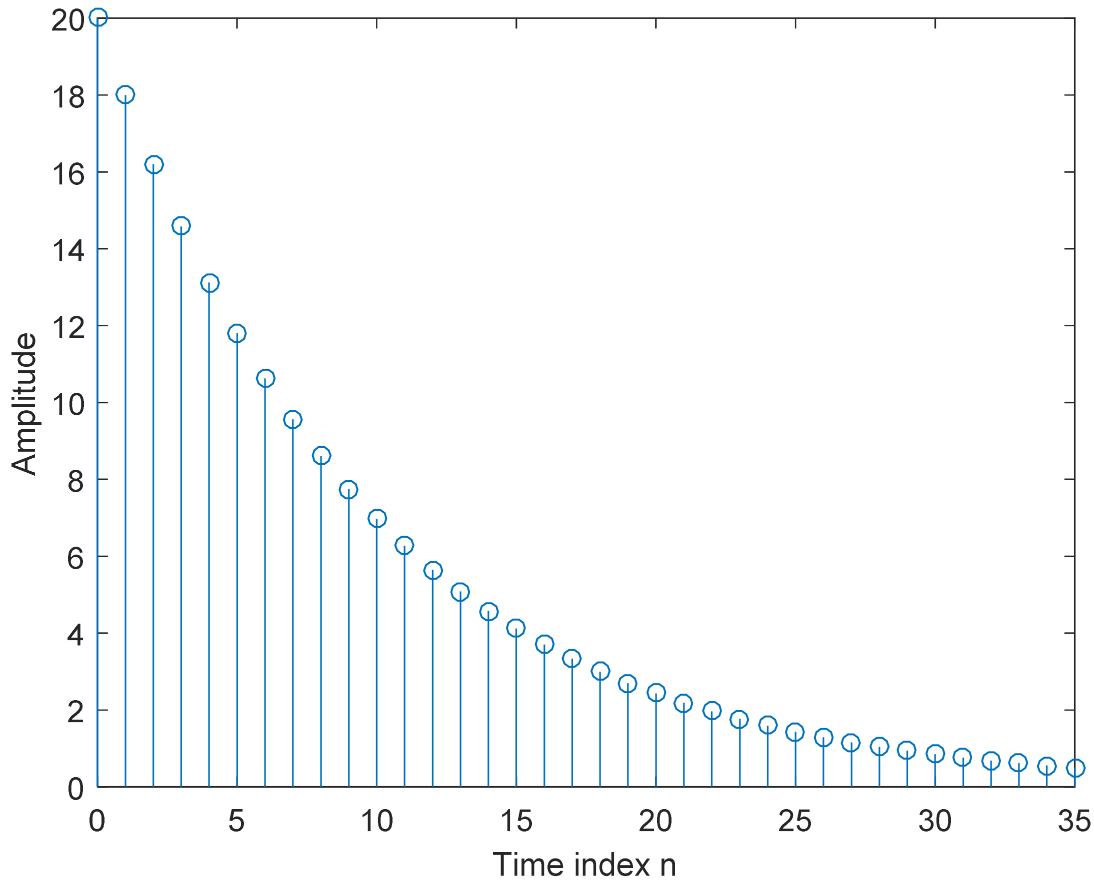
Q1.12 The parameter controlling the rate of growth or decay of this sequence is ***parameter "a".***

The parameter controlling the amplitude of this sequence is ***parameter "K".***

Q1.13 The difference between the arithmetic operators ^ and .^ is:

→ ***If we have a matrix A, then A^2 returns the square of that matrix (the matrix product A*A) while A.^2 returns a matrix in which each element is the square of the corresponding element in A.***

Q1.14 The sequence generated by running Program P1_3 with the parameter a changed to 0.9 and the parameter K changed to 20 is shown below:



Q1.15 The length of this sequence is ***36.***

It is controlled by the following MATLAB command line: `n=0:35;`

It can be changed to generate sequences with different lengths as follows (give an example command line and the corresponding length): `n = 0:199;` ***and now the length is 200.***

Q1.16 The energies of the real-valued exponential sequences x[n] generated in Q1.11 and Q1.14 and computed using the command sum are :

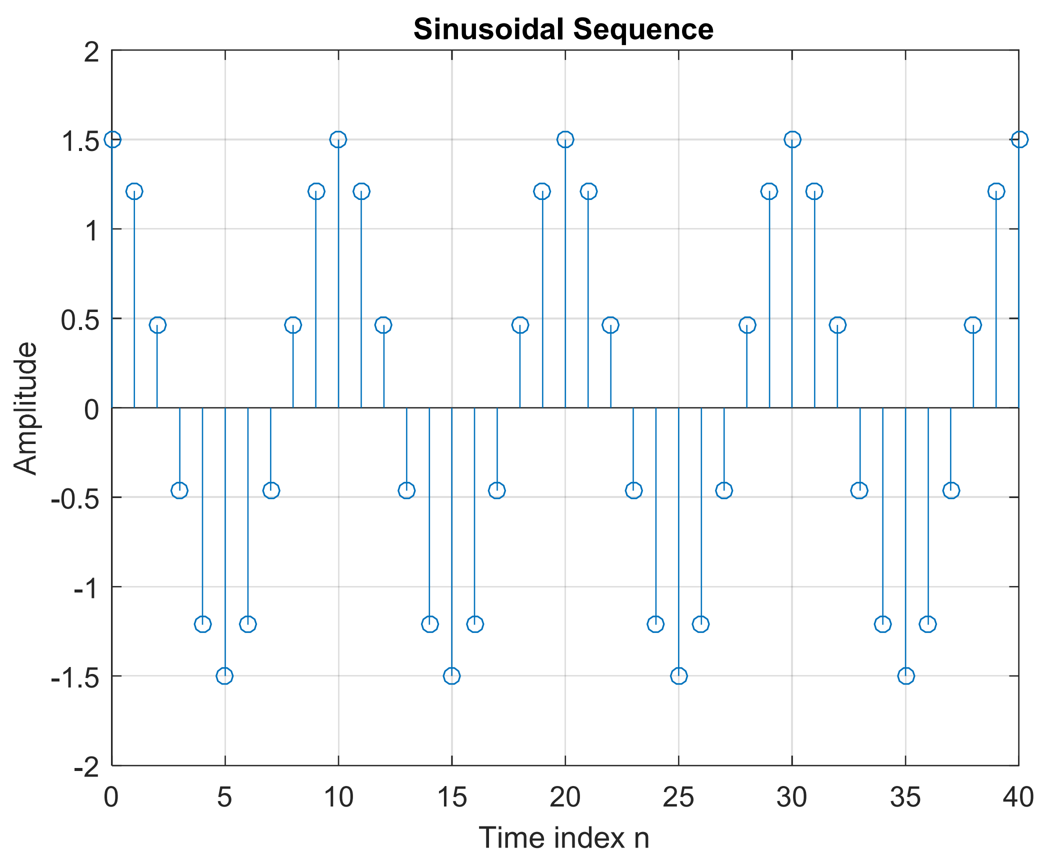
- ***We have:*** *`s=sum(x.^2)`* *so, the energies of the exponential sequences x[n] generated in **Q1.11** is **4.5673e+04** and **2.1042e+03** for that of **Q.14.***

Project 1.3 Sinusoidal sequences

A copy of Program P1_4 is given below.

```
% Generation of a sinusoidal sequence
clc; clear all; close all;
n = 0:40;
f = 0.1;
phase = 0;
A = 1.5;
arg = 2*pi*f*n - phase;
x = A*cos(arg);
stem(n,x);        % Plot the generated sequence
axis([0 40 -2 2]);
grid;
title('Sinusoidal Sequence');
xlabel('Time index n');
ylabel('Amplitude');
axis;
```

Q1.17 The sinusoidal sequence generated by running Program P1_4 is displayed below.



Q1.18 The frequency of this sequence is $f = 0.1\ Hz$.

It is controlled by the following MATLAB command line: `f=0.1;`

A sequence with new frequency ***1Hz*** can be generated by the following command line: `f=1;`

The parameter controlling the phase of this sequence is `phase`.

The parameter controlling the amplitude of this sequence is `A`.

The period of this sequence is $T = \frac{2\pi}{\omega} = \frac{1}{f} = \frac{1}{0.1} = 10$.

Q1.19 The length of this sequence is `41 samples`.

It is controlled by the following MATLAB command line: `n=0:40;`

A sequence with new length `77` can be generated by the following command line: `n=0:76;`

Q1.20 The average power of the generated sinusoidal sequence is :

With period $T = 10$***, we have average power of this sequence***

```
sum(x(1:10).*x(1:10))/10=1.125 (power unit)
```

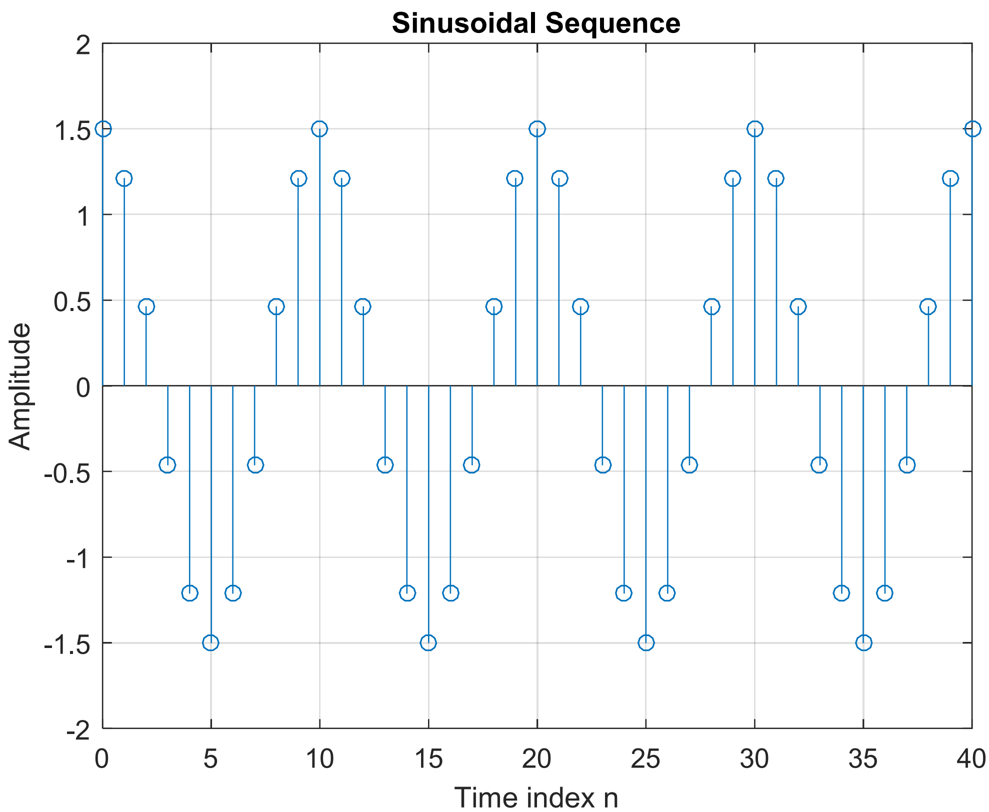
Q1.21 The purpose of `axis` command is used to ***set axis limits and appearance.***

The purpose of `grid` command is used to ***display or hide axes grid lines.***

Q1.22 The modified Program P1_4 to generate a sinusoidal sequence of frequency 0.9 is given below along with the sequence generated by running it.

```
% Generation of a sinusoidal sequence
clc; clear all; close all;
n = 0:40;
f = 0.9;
phase = 0;
A = 1.5;
arg = 2*pi*f*n - phase;
x = A*cos(arg);
stem(n,x);      % Plot the generated sequence
axis([0 40 -2 2]);
grid;
```

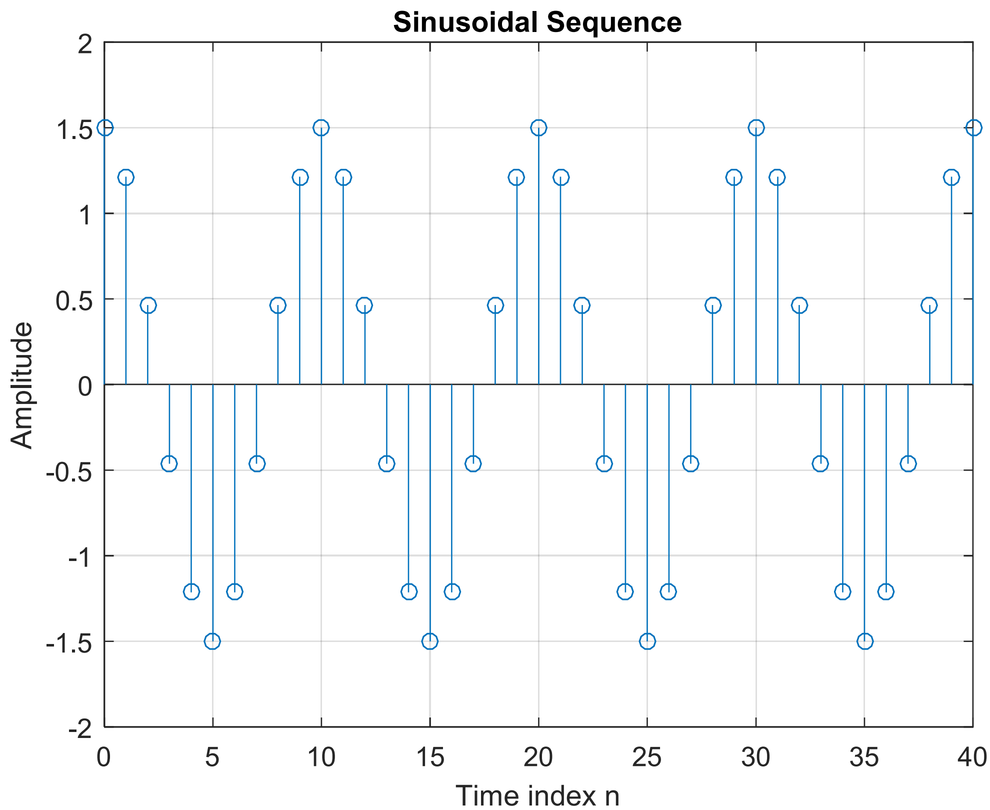
```
title('Sinusoidal Sequence');
xlabel('Time index n');
ylabel('Amplitude');
axis;
```



+ A comparison of this new sequence with the one generated in ***Question Q1.17*** shows that

The two signals have similar shape and both have properties of cosine functions. *As we know, the sine function is a periodic function with period* $T = 2\pi$ *and is even function* $\cos(\omega) = \cos(-\omega)$ *(*). So* $0,9.2\pi - 2\pi = -0,1.2\pi$*, following (*) we get this conclusion.*

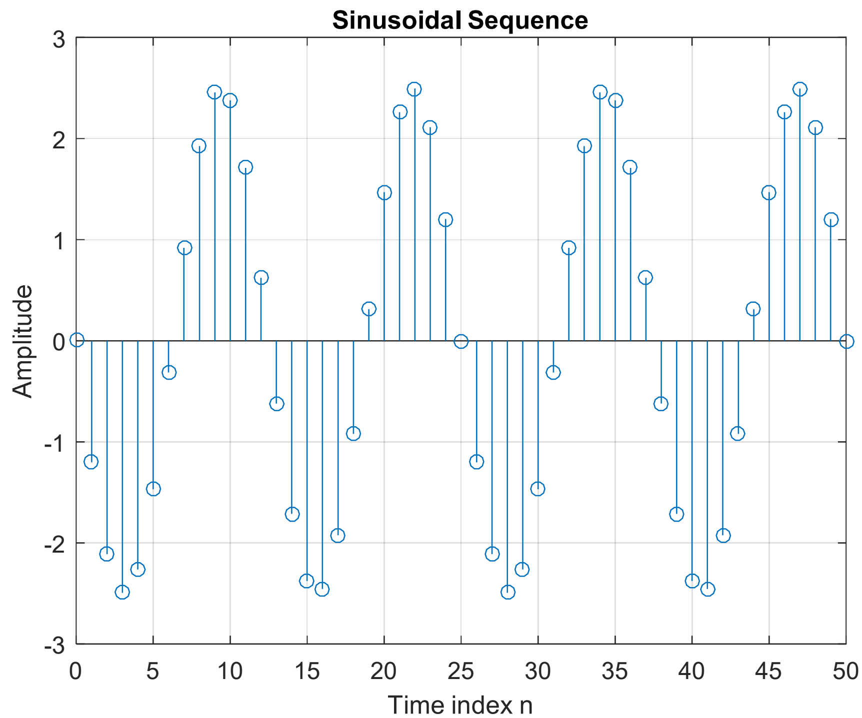
+ A sinusoidal sequence of ***frequency 1.1*** generated by modifying Program P1_4 is shown below.



+ A comparison of this new sequence with the one generated in ***Question Q1.17*** shows that:

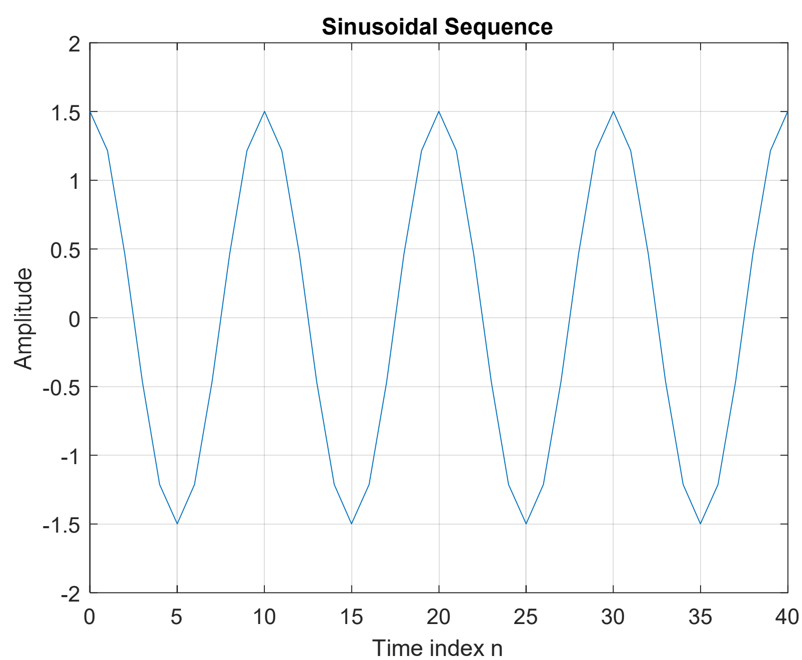
The two signals have similar shape and both have properties of cosine functions one morw time. *As mentioned previously,* ***So*** $1{,}1.2\pi - 2\pi = 0{,}1.2\pi$ *, following (*) we get this conclusion again.*

Q1.23 The sinusoidal sequence of length 50, frequency 0.08, amplitude 2.5, and phase shift of 90 degrees generated by modifying Program P1_4 is displayed below.



The period of this sequence is $T = \dfrac{k2\pi}{\omega} = \dfrac{k}{f} = \dfrac{k}{0.08} = 12.5k = 25,\ 50,\ \ldots.$

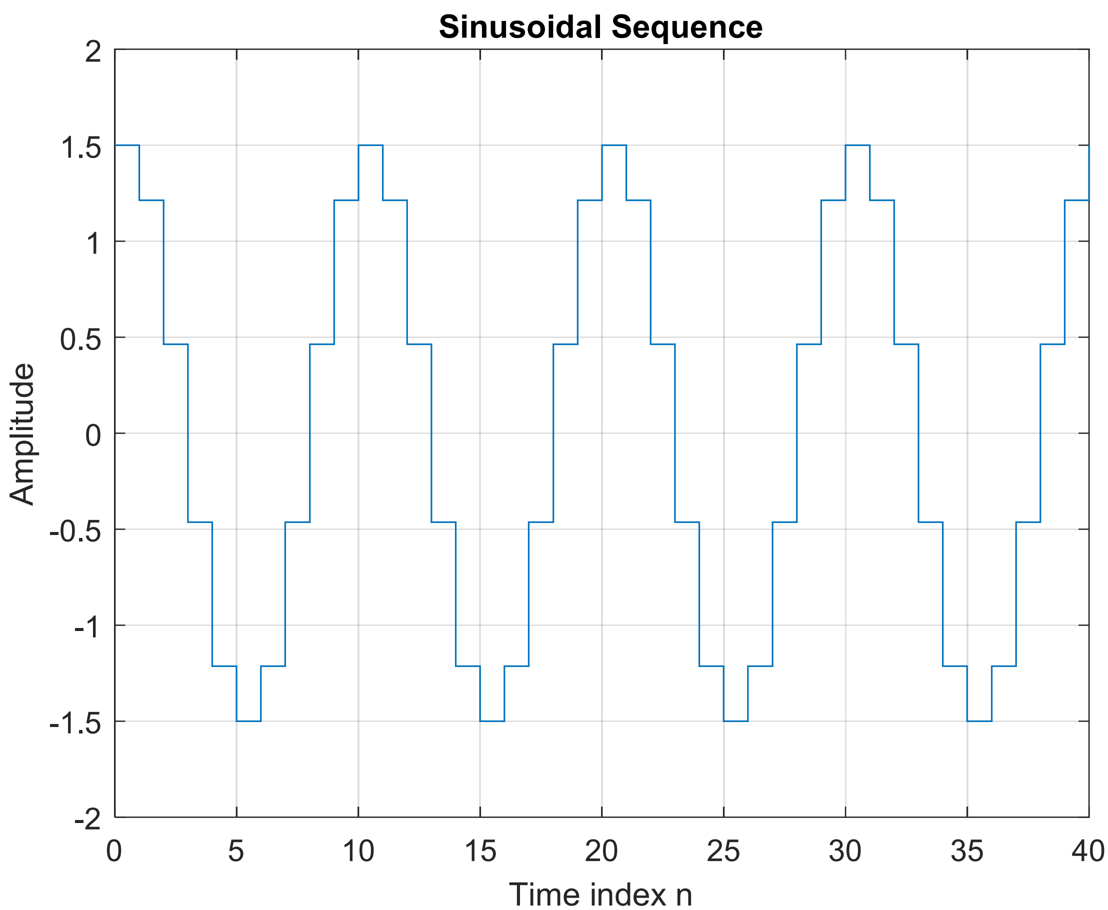
Q1.24 By replacing the stem command in Program P1_4 with the plot command, the plot obtained is as shown below:



→ The difference between the new plot and the one generated in Question Q1.17 is:

The main point of difference between the two is that `plot` ***displays the continuous values for the curve by connecting the points with straight line segments, which approximates the graph of a continuous-time cosine signal. On the other hand,*** `stem` ***displays the discrete values of the points on the curve.***

Q1.25 By replacing the stem command in Program P1_4 with the stairs command the plot obtained is as shown below:



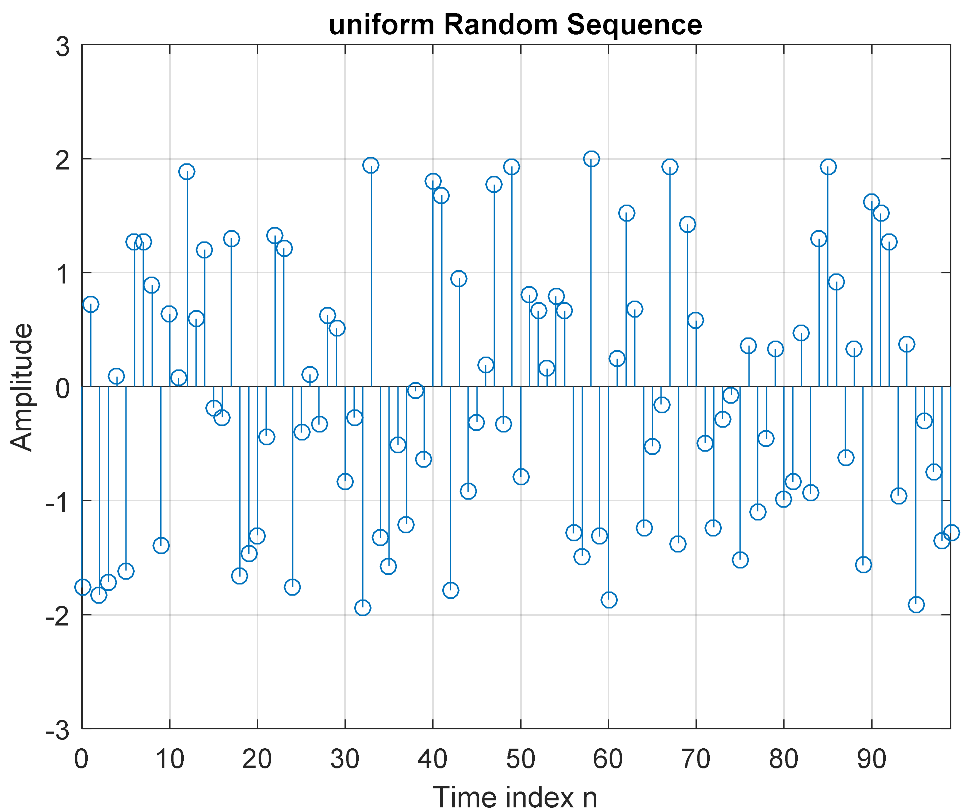
The difference between the new plot and those generated in ***Questions Q1.17*** and ***Q1.24*** is:

→ ***Unlike*** `plot` ***and*** `stem` ***command,*** `stairs` ***command draws a stairstep graph.***

Project 1.4 Random signals

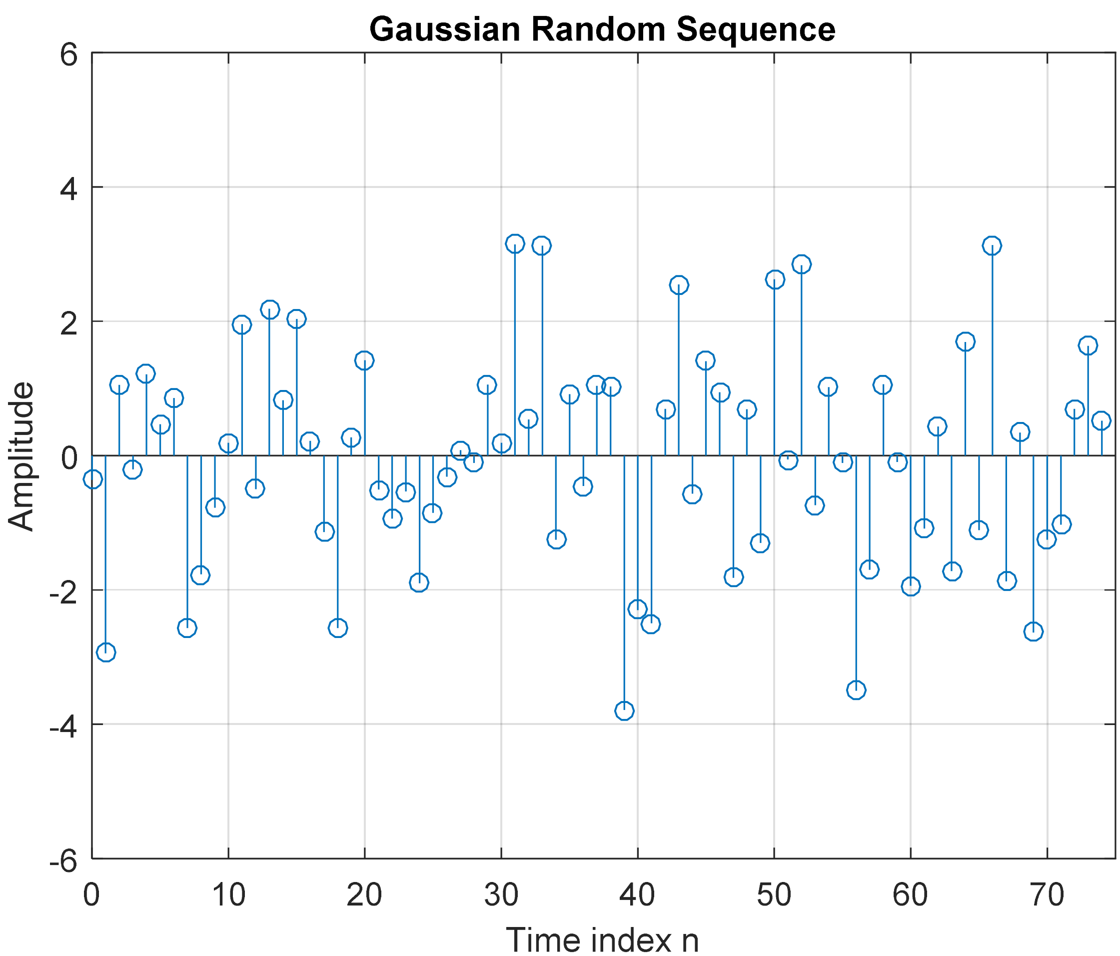
Q1.26 The MATLAB program to generate and display a random signal of length 100 with elements ***uniformly distributed*** in the interval [–2, 2] is given below along with the plot of the random sequence generated by running the program:

```
n=0:99;
x = 4*(rand(1,100)-0.5); %4*([0,1]-0.50=[2,2]
clf;
stem(n,x);
title('uniform Random Sequence');
xlabel('Time index n');
ylabel('Amplitude');
axis([0 99 -3 3]);
grid on
```



Q1.27 The MATLAB program to generate and display a ***Gaussian random signal*** of length 75 with elements normally distributed with zero mean and a variance of 3 is given below along with the plot of the random sequence generated by running the program:

```
clc; clear all; close all;
n=0:74;
mean=0;
dlc=sqrt(3);
x=dlc*randn(1,75)+mean;   %p=aX+b
stem(n,x);
axis([0 75 -6 6]);
grid;
title('Gaussian Random Sequence');
xlabel('Time index n');
ylabel('Amplitude');
```

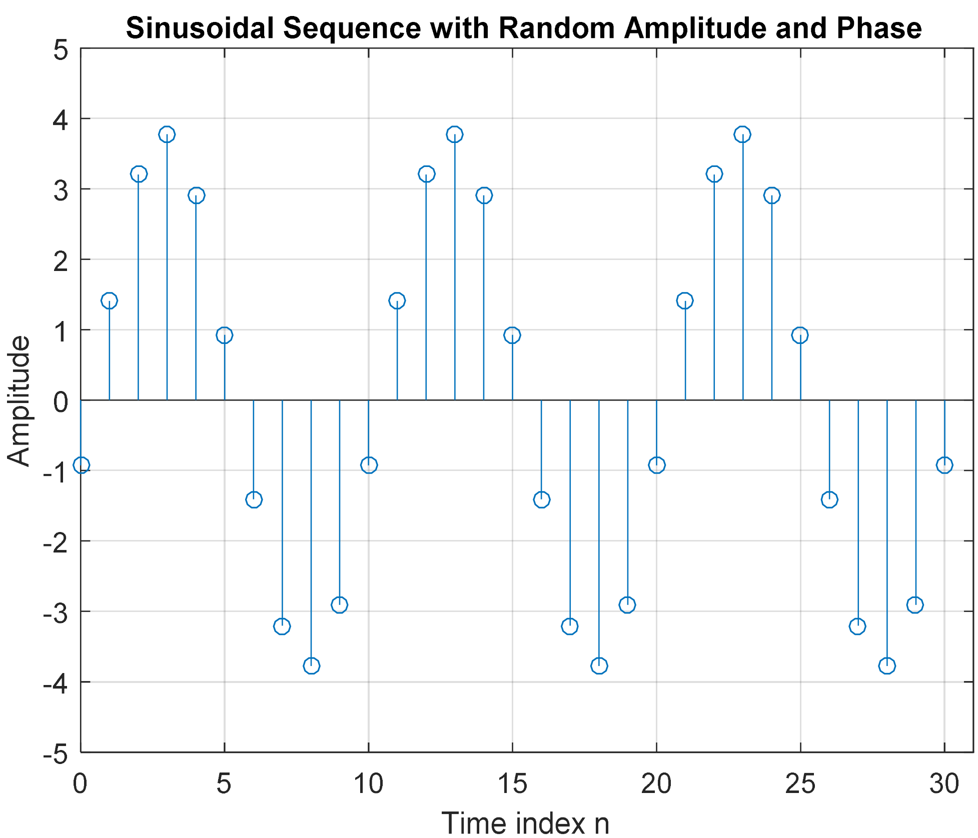


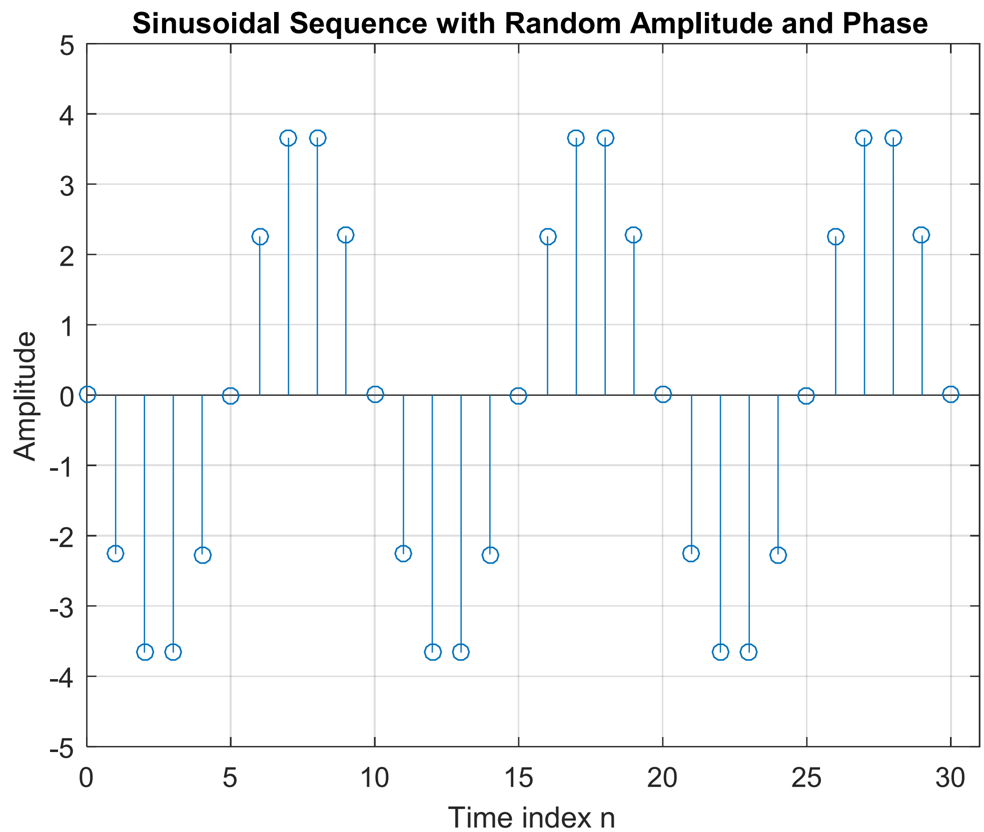
Q1.28 The MATLAB program to generate and display five sample sequences of a random sinusoidal signal of length 31 $\{x(n) = A\cos(\omega_0 n + \phi)\}$

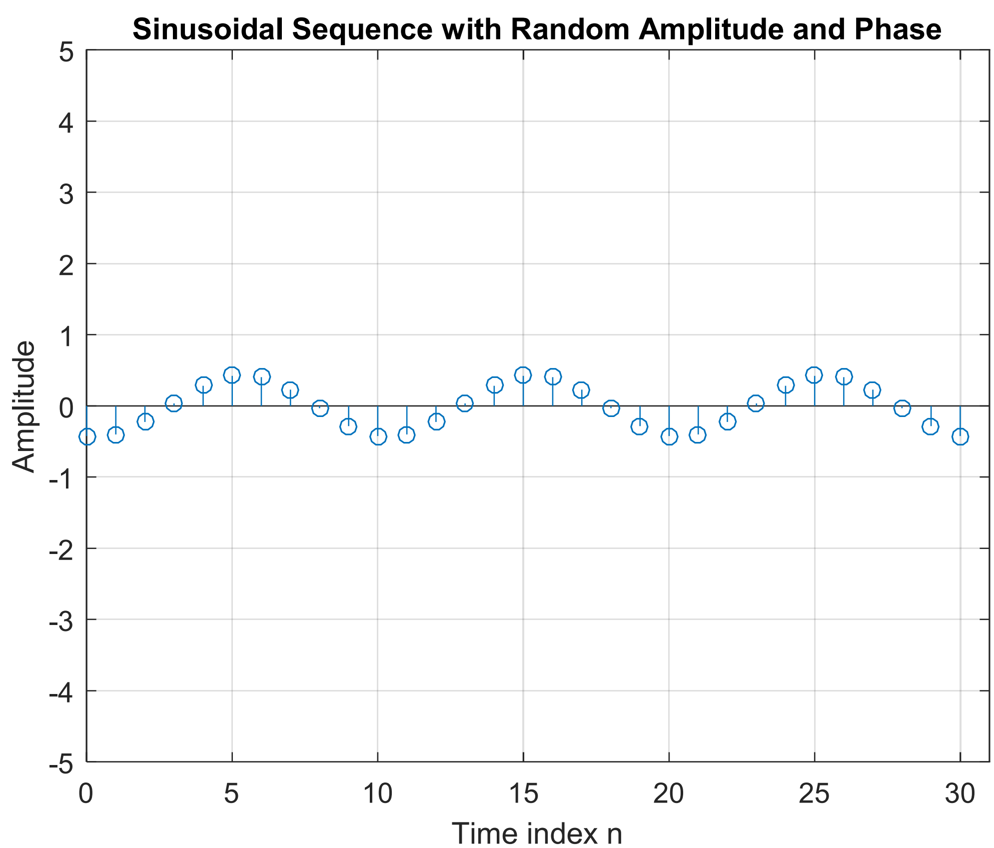
where the amplitude A and the phase ϕ are statistically independent random variables with uniform probability distribution in the range $0 \le A \le 4$ for the amplitude and in the range

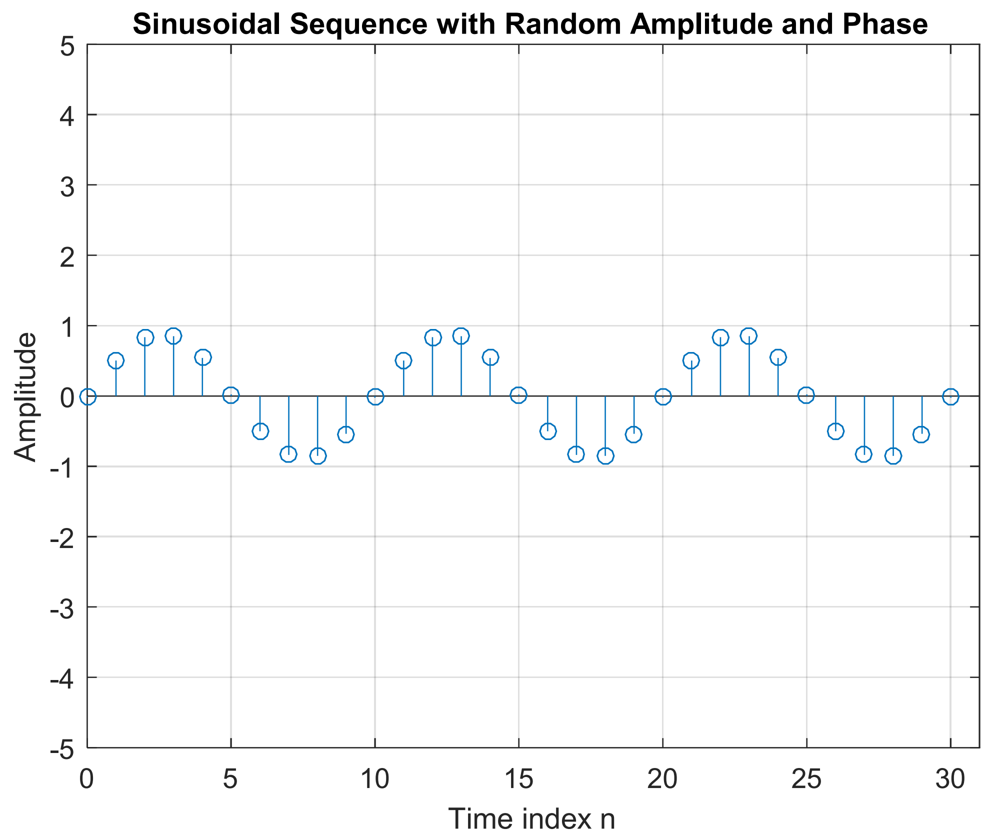
$0 \le \phi \le 2\pi$ for the phase is given below. Also shown are five sample sequences generated by running this program five different times.

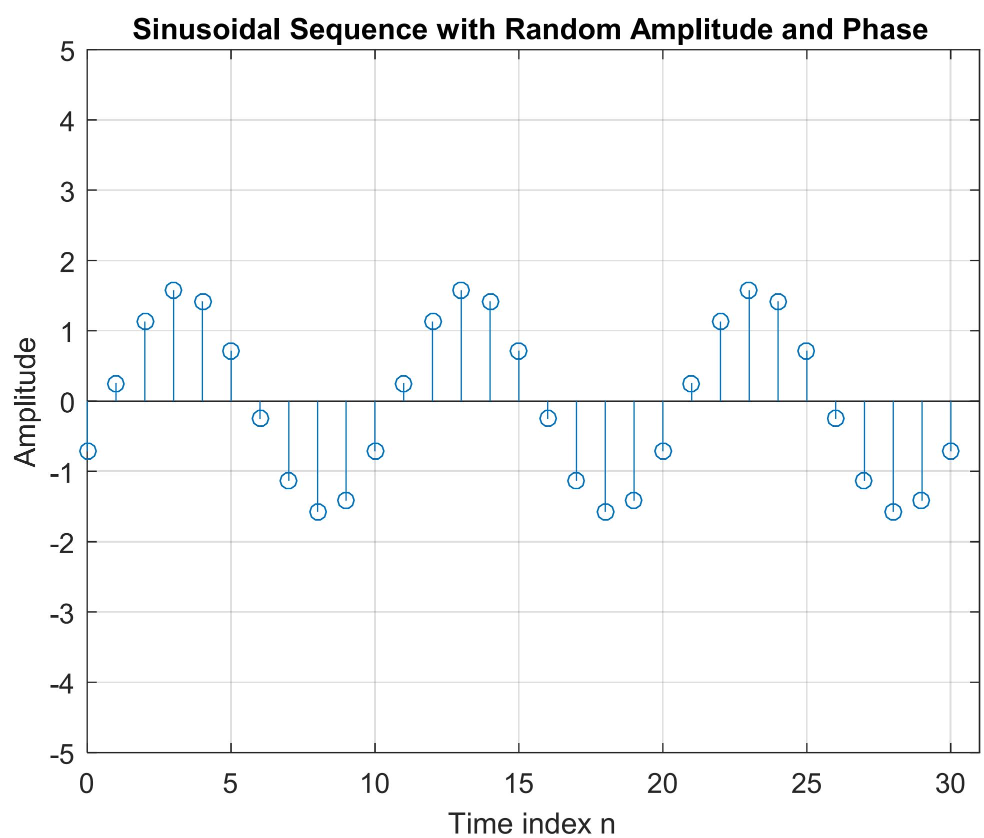
```
clc; clear all; close all;
n = 0:30;
f = 0.1;
Amax = 4;
phimax = 2*pi;
A = Amax*rand;
phi = phimax*rand;
arg = 2*pi*f*n + phi;
x = A*cos(arg);
stem(n,x);
axis([0 31 -5 5]);
grid;
title('Sinusoidal Sequence with Random Amplitude and
Phase');
xlabel('Time index n');
ylabel('Amplitude'); axis;
```




Sinusoidal Sequence with Random Amplitude and Phase
Amplitude
Time index n


Sinusoidal Sequence with Random Amplitude and Phase
Amplitude
Time index n


Sinusoidal Sequence with Random Amplitude and Phase
Amplitude
Time index n
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
5
10
15
20
25
30


Sinusoidal Sequence with Random Amplitude and Phase
Amplitude
Time index n
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
5
10
15
20
25
30

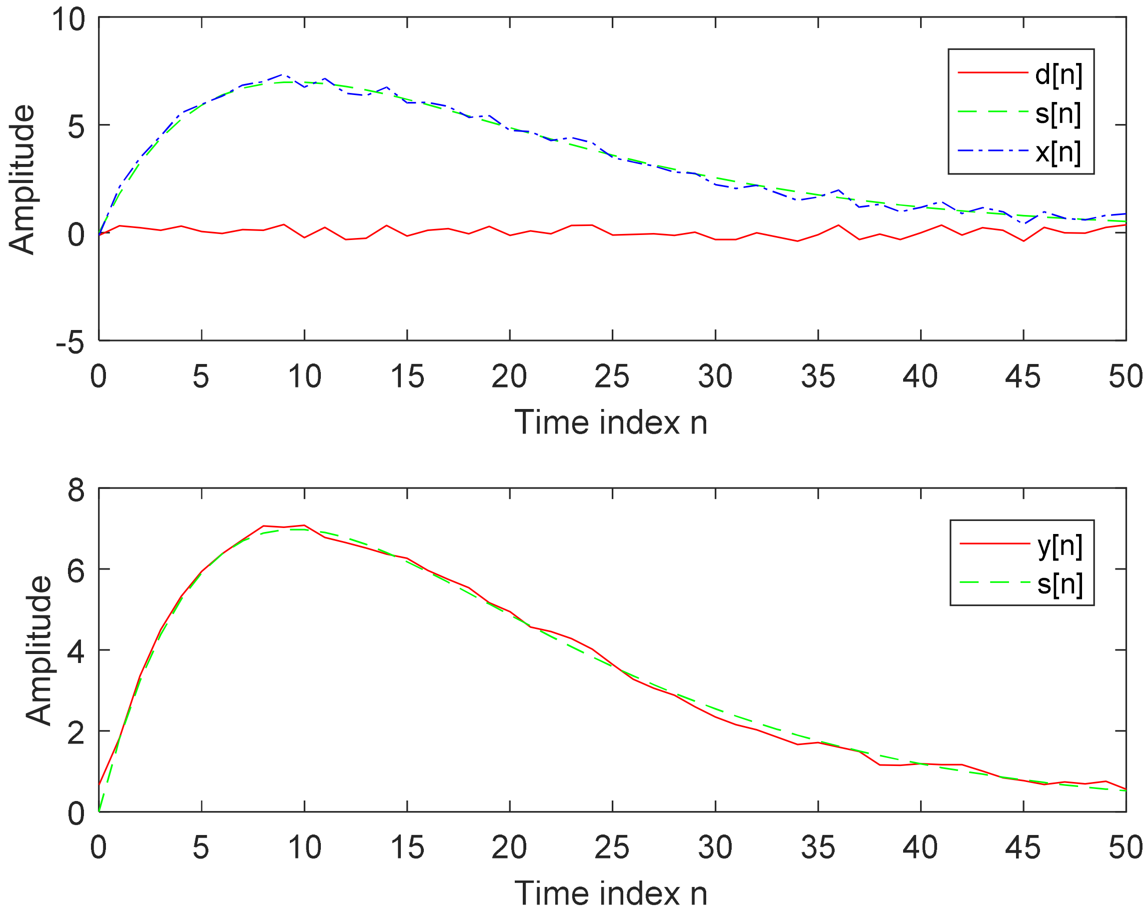
1.2 SIMPLE OPERATIONS ON SEQUENCES

Project 1.5 Signal Smoothing

A copy of Program P1_5 is given below.

```
% Signal Smoothing by Averaging
clc; clear all; close all;
R = 51;
d = 0.8*(rand(R,1) - 0.5); % Generate random
noise
m = 0:R-1;
s = 2*m.*(0.9.^m); % Generate uncorrupted
signal
x = s + d'; % Generate noise corrupted
signal
subplot(2,1,1);
plot(m,d','r-',m,s,'g--',m,x,'b-.');
xlabel('Time index n');
ylabel('Amplitude');
legend('d[n] ','s[n] ','x[n] ');
x1 = [0 0 x];x2 = [0 x 0];x3 = [x 0 0];
y = (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1),'r-',m,s,'g--');
legend( 'y[n] ','s[n] ');
xlabel('Time index n');
ylabel('Amplitude');
```

Q1.29 The signals generated by running Program P1_5 are displayed below:



Q1.30 The uncorrupted signal s[n] is ***a decreasing exponential function.***

The additive noise d[n] is ***a random sequence uniformly distributed between -0.4 and + 0.4.***

Q1.31 The statement x = s + d ***CANNOT*** be used to generate the noise corrupted signal because ***d is a column vector, whereas s is a row vector; it is necessary to transpose one of these vectors before adding them.***

Q1.32 The relations between the signals x1, x2, and x3, and the signal x are ***all of those signals x1, x2, and x3 are another versions of x, with one is at the left and one is at the right. The signal x1 is a delayed version of x, the signal x2 is equal to x and x3 is a time advanced version of x.***

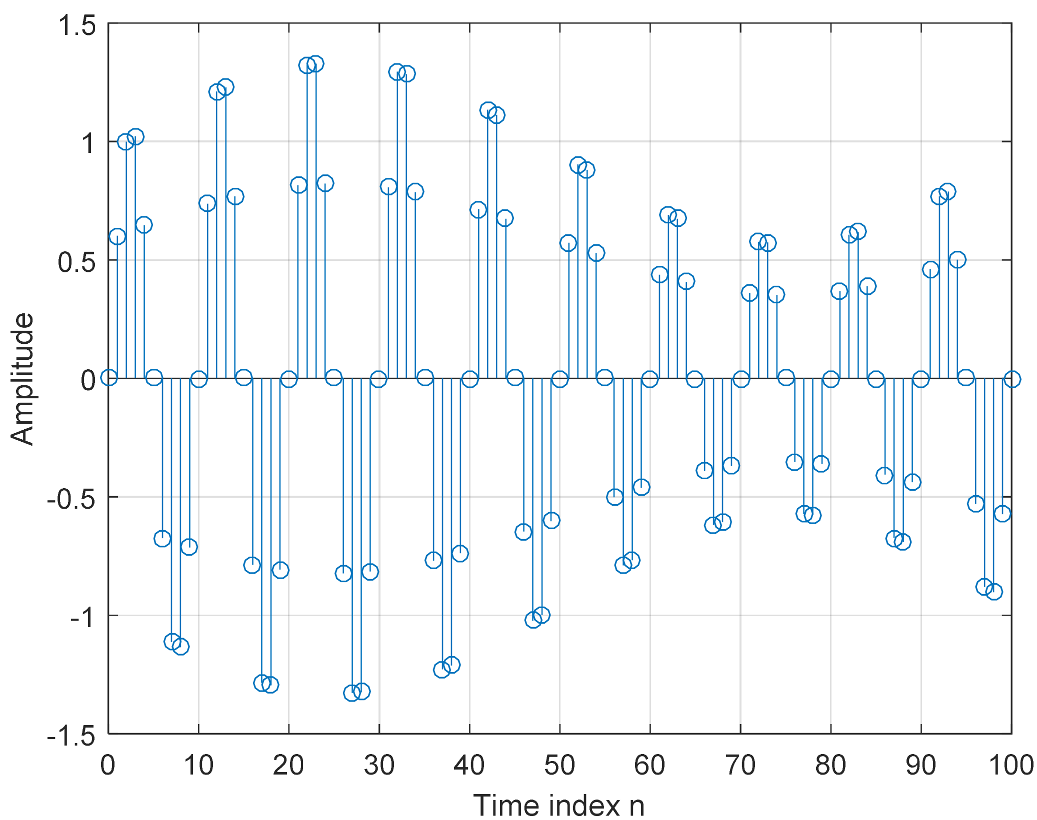
Q1.33 The purpose of the `legend` command is ***used to add legend to graph.***

Project 1.6 Generation of Complex Signals

A copy of Program P1_6 is given below.

```
% Generation of amplitude modulated sequence
clc; clear all; close all;
n = 0:100;
m = 0.4;fH = 0.1; fL = 0.01;
xH = sin(2*pi*fH*n);
xL = sin(2*pi*fL*n);
y = (1+m*xL).*xH;
stem(n,y);grid;
xlabel('Time index n');ylabel('Amplitude');
```

Q1.34 The amplitude modulated signals y[n] generated by running Program P1_6 for various values of the frequencies of the carrier signal xH[n] and the modulating signal xL[n], and various values of the modulation index m are shown below:

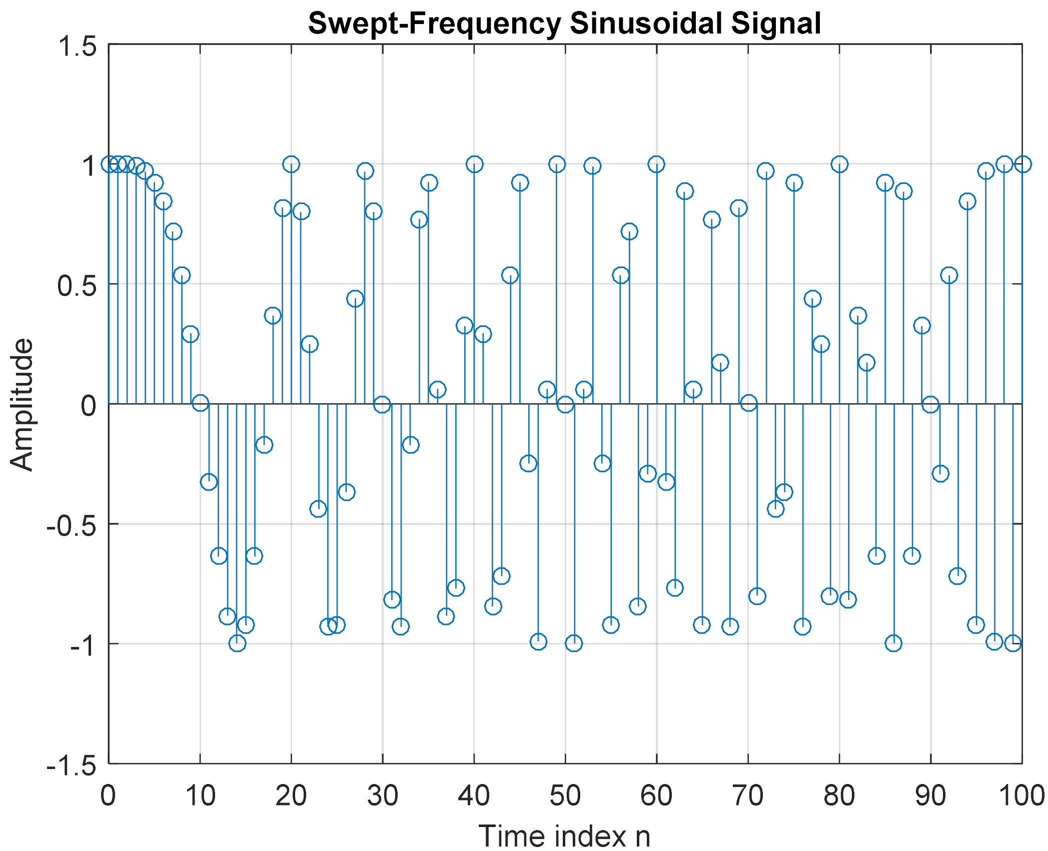


Q1.35 The difference between the arithmetic operators * and .* is : ***"*" multiplies two conformable matrices or vectors using matrix multiplication. ".*" takes the pointwise products of the elements of two matrices that have the same dimensions.***

A copy of Program P1_7 is given below.

```
% Program P1_7
clc; clear all; close all;
n = 0:100;
a = pi/2/100;
b = 0;
arg = a*n.*n + b*n;
x = cos(arg);
stem(n, x);
axis([0,100,-1.5,1.5]);
title('Swept-Frequency Sinusoidal
Signal');
xlabel('Time index n');
ylabel('Amplitude');
grid;
axis;
```

Q1.36 The swept-frequency sinusoidal sequence x[n] generated by running Program P1_7 is displayed below.



Q1.37 The minimum and maximum frequencies of this signal are:

As the frequency of a sinusoidal signal is the derivative of its phase with respect to time, so we have $\omega = \left(an^2 + bn\right)' = 2an + b = 2.\dfrac{\pi}{2.100}.n$

\+ *The minimum frequencie is 0 when* $n = 0 \Rightarrow \omega = 0$

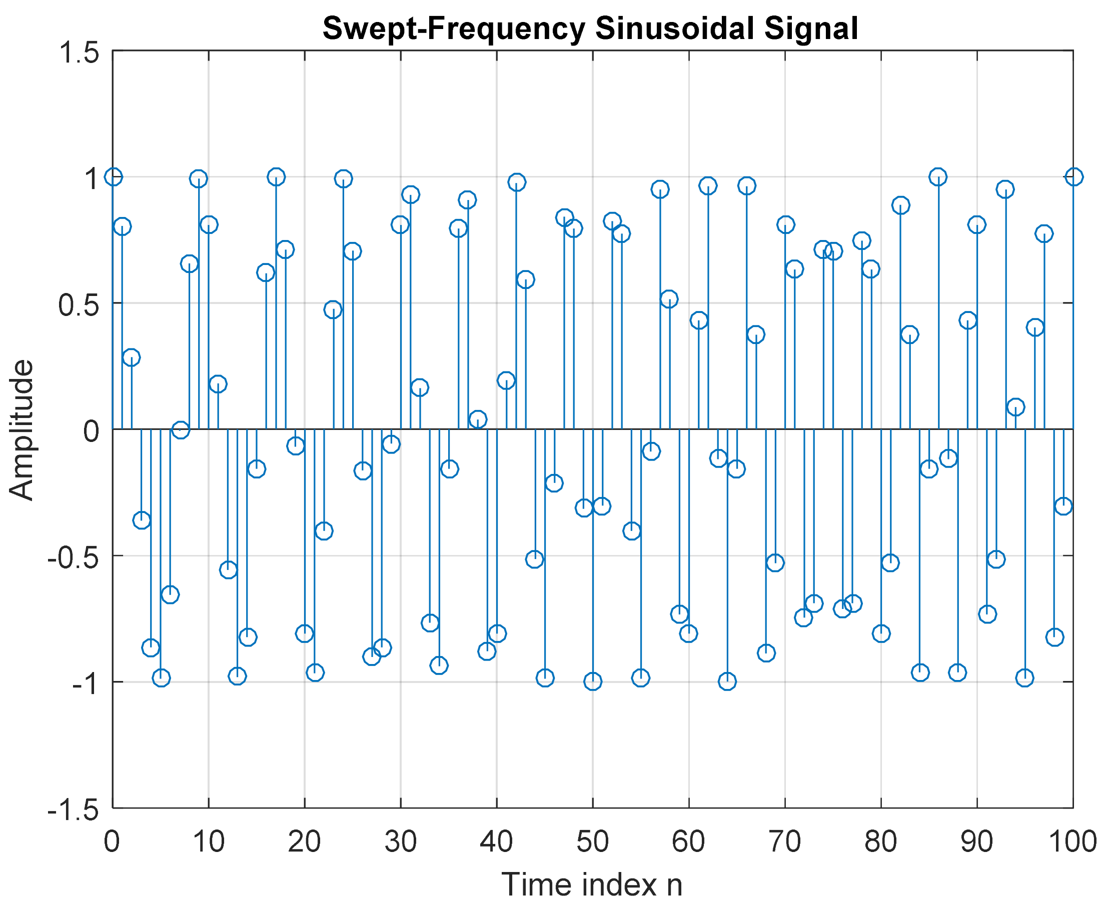
\+ *The maximum frequencie is* 0.5 *when* $n = 100 \Rightarrow \omega = 2.\dfrac{\pi}{2.100}.100 = \pi$

Q1.38 The Program 1_7 modified to generate a swept sinusoidal signal with a minimum frequency of 0.1 and a maximum frequency of 0.3 is given below:

We have the following system of equations:

$$\begin{cases} \omega_{\min(n=0)} & = 2.a.0 + b = 0{,}1.2\pi \\ \omega_{\max(n=100)} & = 2.100.a + b = 0{,}3.2\pi \end{cases} \Rightarrow \begin{cases} b = 0.2\pi \\ a = 2\pi.10^{-3} \end{cases}$$

```
clc; clear all; close all;
n = 0:100;
a = 2*pi*10^(-3);
b = 0.2*pi;
arg = a*n.*n + b*n;
x = cos(arg);
stem(n, x);
axis([0,100,-1.5,1.5]);
title('Swept-Frequency Sinusoidal Signal');
xlabel('Time index n');
ylabel('Amplitude');
grid;
```



1.3 WORKSPACE INFORMATION

Q1.39 The information displayed in the command window as a result of the `who` command is ***a list of variables in current workspace.***

Q1.40 The information displayed in the command window as a result of the `whos` command is ***a list of variables in workspace, with sizes and types.***

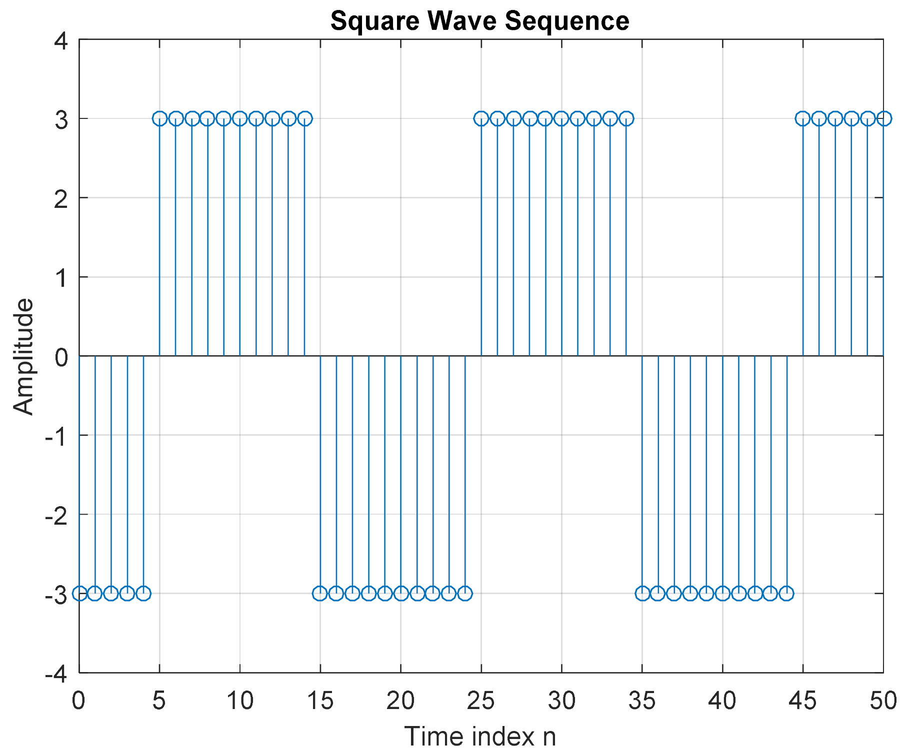
1.4 OTHER TYPES OF SIGNALS (Optional)

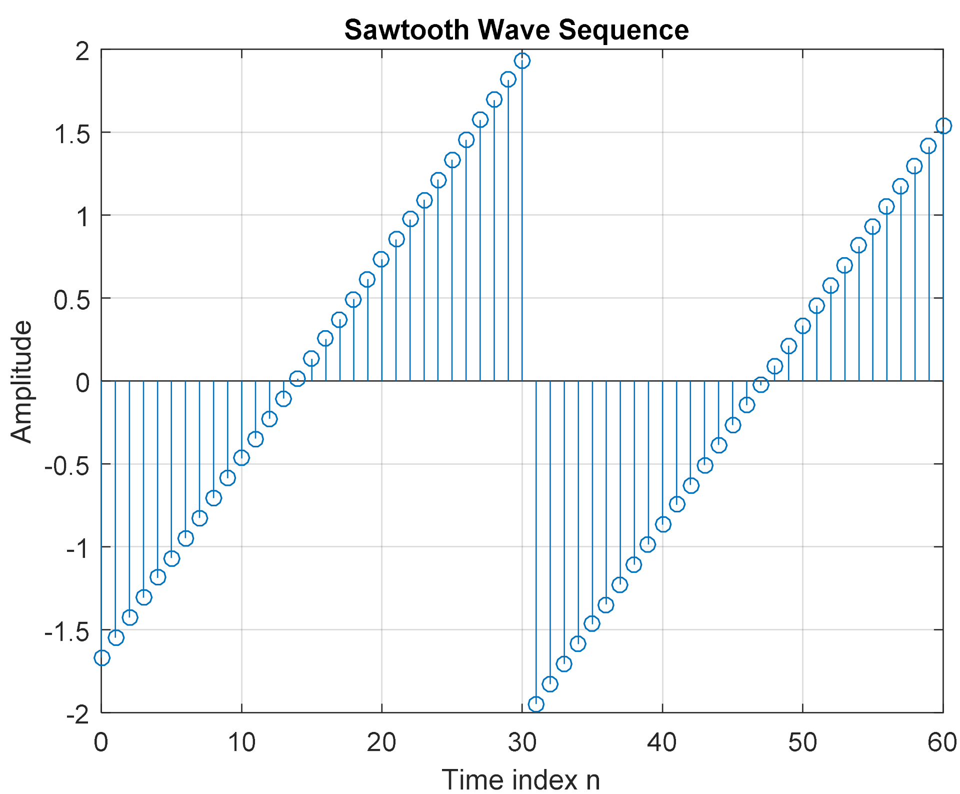
Project 1.8 Squarewave and Sawtooth Signals

Q1.41 MATLAB programs to generate the square-wave and the sawtooth wave sequences of the type shown in Figures 1.1 and 1.2 are given below along with the sequences generated by running these programs:

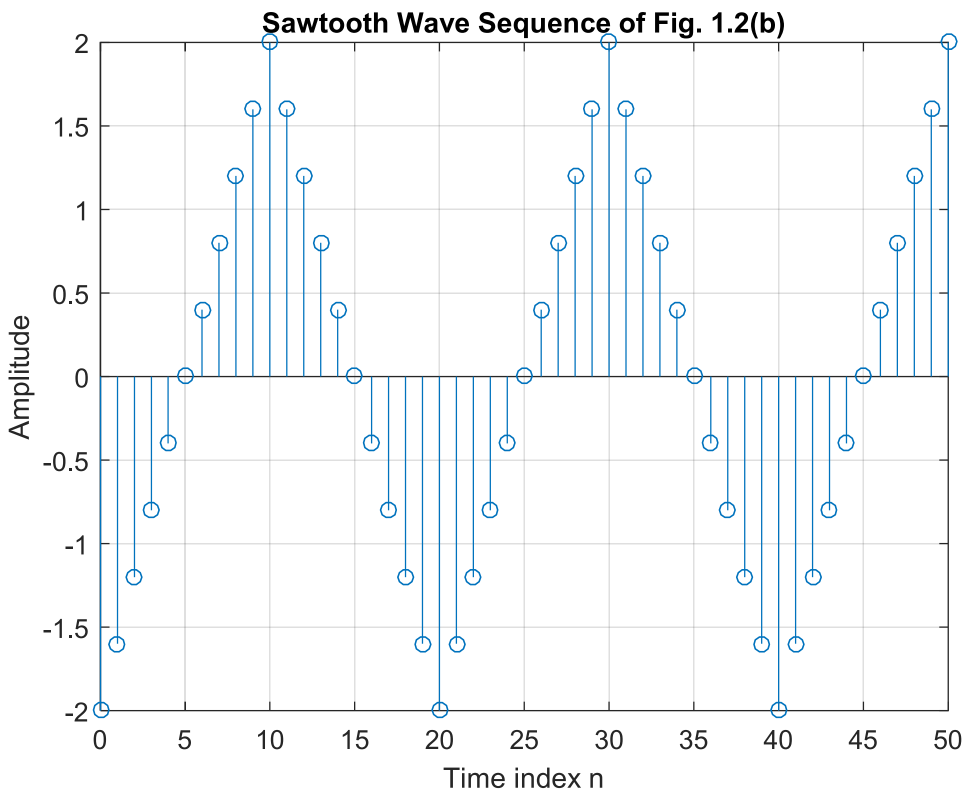
```
clc; clear all; close all;
n = 0:50;
f = 0.05;
phase = -pi/2;
duty=50;
A = 3;
arg = 2*pi*f*n + phase;
x = A*square(arg,duty);stem(n,x); % Plot the
generated sequence
axis([0 50 -4 4]);
grid;
title('Square Wave Sequence');
xlabel('Time index n');
ylabel('Amplitude');
axis;
n = 0:60;
f = 0.03;
phase = pi/6;

clc; clear all; close all;
peak = 1;
A = 2.0;
arg = 2*pi*f*n + phase;
x = A*sawtooth(arg,peak);
stem(n,x); % Plot the generated sequence
axis([0 60 -2 2]);
grid;
title('Sawtooth Wave Sequence ');
xlabel('Time index n');
ylabel('Amplitude');
```


Square Wave Sequence
4
3
2
1
0
-1
-2
-3
-4
Amplitude
0
5
10
15
20
25
30
35
40
45
50
Time index n


Sawtooth Wave Sequence
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Amplitude
0
10
20
30
40
50
60
Time index n

```
clc; clear all; close all;
n = 0:50;
f = 0.05;
phase = 0;
peak = 0.5;
A = 2.0;
arg = 2*pi*f*n + phase;
x = A*sawtooth(arg,peak);
clf; % Clear old graph
stem(n,x); % Plot the generated sequence
axis([0 50 -2 2]);
grid;
title('Sawtooth Wave Sequence of Fig. 1.2(b)');
xlabel('Time index n');
ylabel('Amplitude');
axis;
```



Date: 29/08/2023 **Signature: Do Trung Hau**