Experiment No1

Aim: Signal generation

Activity1: Generate 100 samples of the following functions and plot using the stem and **plot** function

a) X (t) = 4 *cos (5π t-π /4),* the time interval should be, M = 0.01

b) X[n] = 4 cos (n π)

c) X (t) = cos (4t) + 2sin (8t), the time interval should be Δt = 0.1

d) X (t) = $5e^{-j\omega t}$, for -1≤ t ≤ 1, Δt = 0.01, -10≤ ω≤ 10,

Activity2: Write MATLAB code to plot the signals given bellow. Scale your time axis so that a sufficient amount of the signal is being plotted. Use subplot to give 4 plots per page; label your plots with 'Time (sec)' on the x-axis for the continuous time signals and 'n' for discrete time signals. The y-axis should be labeled 'x (t)' or 'x[n]'; the title should be the problem number, for example 'a)'.

(a) X (t) = 0 if t< -4
= t+2 if -4 ≤ t < 3
= t- 2 if 3 ≤ t

(b) X (n) = 0 if n < 2
= 2n – 4 if 2 ≤ n < 4
= 4- n if 4 ≤ n

Experiment NO.2

Aim: To create user function for generating delta-function, unit-step function and periodic signal.

Impseq (n0, nl, n2) % to generate an impulse sequence with length (n 1:n2) and delay n0

Stepseq (n0, n1, n2) % to generate a step sequence with length (n1:n2) and delay n0

periodic(x,p) % periodic extension of x for p times (periods)

Activity 1: Write a function to generate the delta function 8(n) MATLAB Program:

```
Function [x, n] =impseq (n0, n1, n2)
%generates a delta function δ(n) =delta (n-n0); nl <=n0<=n2
% x is the delta function
%-----------------------------------------------
%[x, n] =impseq (n0, n 1, n2)
% no=delay or the location where impulse sequence exists n=[nl: n2];
x= [(n-no) = =0];
```

Exercise:

Use the delta-function to generate the following signals and plot using stem function

(a) $x(n) = \delta(n) + 2\ \delta\ (n-2) + 3\ \delta\ (n+3)$

(b) $x(n) = \delta\ (n) + 2\ \delta\ (n-2) + 2\ \delta\ 8(n+2) + 3\ \delta\ (n-3) + 3\ \delta\ 8(n+3)$

Activity 2: Write a function to generate the unit-step function u (n) MATLAB Program:

```
Function [x, n] =stepseq (n0, nl, n2)
%to generate a step sequence x (n) =u (n-n0); nl <=n0<=n2
%  -------------------------------------------------------------
%[x, n] =stepseq (n0, nl, n2)
n= [nl: n2];
x= [(n-n0>=0];
```

Exercise:

Use the step-function to generate the following signals and plot using stem function.

(a) X (n) = u (n-2) – u (n-8)

(b) X (n) = n^2 [u (n + 5) - u (n - 6)] + 1 05(n) + 20(0.5)" [u (n - 4) - u (n -10)]

Activity 3: Write a function to generate the unit-step function u (n) MAT LAB Program:

```
Function [y] = periodic(x, p)
% periodic extension of the signal x (n)
 % y=periodic(x, p)
%-----------------------------------------
% x=finite duration signal %
p=no. of periods
% y=periodic extension of the signal x(n)
xtidle=x'*ones( 1,p);
xtidle=xtidle (: );
y=xtidle';
```

Exercise:

1. (a)x = {1, 2, 3, 4, 5}. Plot four period.

   (b)x = { .... 1,2,3,2,1,2,3,2,1, ... }*periodic.* Plot 5 periods.

Q.2. Let *x (n)* = [1,-2, 4, 6, -5, 8, 1] Generate and plot samples (use stem function) of the following sequences.

(a) *X (n)* = 3x (n + 2) + x (n - 4) - 2x (n)

(b) *X (n)* = x (n + 4) x (n -1) + x (2 - n) x (n)

Experiment No.3

AIM: To create user f~mction for performing signal operation: folding, Shifting, signal addition

Activity 1: to create user function to fold any given signal MATLAB Program

---

```
Function [y, n] =sigfold(x, n);
% implements y (n) =x (-n)
%---------------------------
%[y, n] =sigfold(x, n) %
y=fliplr(x);
n= -fliplr (n);
```

Exercise:

Generate the and plot the following signal using the function sigfold

y(n)= x(n)+x(-n), where x(n)= *[-1,2,3,-2,-3,5,6]*

Activity 2: To create the user function to time-shift any given signal MATLAB Program

---

```
Function [y, n] =sigshift(x, m, n0);
%implement y (n) =x (m-n0)
%----------------------------
% [y, n] =sigshift(x, m, n0) n=m+n0;
y=x;
```

Exercise:

Let $x(n) = \{1,1,-2,4,\underset{\uparrow}{6},5,8,10\}$. Generate and plot samples (use stem function) of the following

Sequences.

(a) y (n) = 3x (n+2)

(b) y(n) = x (2-n)

Activity 3: To create the user function for adding two finite duration signal MATLAB Program.

---_ ..

```
Function [y] = sigadd (xl, x2)
% implementes y (n) =x 1 (n) +x2(n)
% _________________________________
% [y] =sigadd (xl, x2)
% y=sum sequence %
xl = first sequence %
x2= first sequence %
[n1x1, n2x 1]=size(x 1);
[nlx2, n2x2]=size(x2);
 nl =min (nlxl, nlx2);
n2=max (n2x 1, n2x2);
yl =[ zeros(1 ,nIx I-nl ),x 1 ,zeros(1 ,n2-n2xl )];
y2=[ zeros(1 ,nl x2-nl ),x2,zeros( 1 ,n2-n2x2)];
y=yl+y2;
```

Exercise:

Let $x(n) = \{1,-2, 4, 6, -1, 5, 8, 10\}$. Generate and plot samples (use stem function) of the following sequences.

(a) $y(n) = 3x(n+2) + x(2-n)$

(a) $y(n) = 3x(n + 2) + x(n - 4) - 2x(n)$

Experiment NO.4

AIM: Linear Time Invariant Systems.

Activity 1:

Linearity: A discrete time system T [.] is linear if and only it satisfy the principle of superposition i.e.

$Y(n) = T[x(n)]$

$Y_1(n) = T[x_1(n)]$; $y_2(n) = T[x_2(n)]$

If $x(n) = aX_l(n) + bX_2(n)$ then

$Y(n) = aYl(n) + bY2(n)$

Exercise:

Two discrete-time systems Tl [.] and *T2 [.]* are defined as *1;*

$T[x(n)] = 3x(n) + 4$

$T_2[x(n)] = x(n) + 2x(n-1) - x(n-2)$

Let $x_l(n) = 10\,(0.2)^n$ and $x_2(n) = 5\cos((0.2\pi n)$, for $0 \le n \le 100$.

Using these sequences, test the linearity of the above systems. Choose any values for constants ***a*** and ***b***.

Activity 2:

Time-invariallce:

If $y(n) = T[x(n)]$

$Y(n, k) = T[x(n-k)]$ % response to a shifted input $x(n-k)$

Y (n-k) % the output sequence *yen)* is shifted with *k*

If $y(n-k) = y(n, k)$ then the system said to be time-invariant

Exercise:

For the system T1 [.] and T2 [.] given above,
Check whether the systems are time invariant or not.

Experiment No.5

Response of LTI Systems

Response or an LTI system to an input *x (n)* is simply the convolution or *X (II)* with impulse response hen). MAT LAB provides a built-in function called conv that compute convolution between two finite duration sequences, The **conv** function assumes that two sequences begins at *n =0* and is invoked by

y= conv (x, h) % x and h are row vectors containing samples of the sequence x (n) and h (n)

However, the conv function neither provides nor accepts any timing infom1ation if the sequences are of arbitrary support.
Modify the conv function, which can perform convolution of any arbitrary support sequences and also provide the timing information of the convolved sequence.

```
function [y, ny] =conv _m(x, nx, h, nh);
%modified convolution function which provide both %the
convolved sequence and its time-duration
%---------------------------------------------
%[y,ny ]=conv _ m(x,nx,h,nh)
%[y,ny]=convoled sequence
%[x,nx]=first sequence
 % [h, nh] =second sequence
nyb=nx (1) +nh (l);
nye=nx (length (x)) +nh (length (h));
ny=nyb: nye;
y=conv(x, h);
```

Exercise

Q.l. Perform the linear convolution to the following pair of sequences:

(a) $h(n) = (-2)^n$ for $-2 \le n \le 10$

$x(n) = e^{-n}$ for $-2 \le n \le 20$

(b) $h(n) = 3^n \cos(\pi n/3)\, u(n)$ $x(n) = 2$ for $-3 \le n \le 20$

Use the stem function to plot the sequence *x (n), h (n)* and the convolved output.

Q.2. Linear convolution follow associative and identity property as

$$[X_1(n) * X_2(n)] * X_3(n) = X_1(n) * [x_2(n) * x_3(n)]$$

$$x(n) * \delta(n - n_0) = x(n - n_0)$$

Using the following three sequences, verify the above properties.

$X1(n) = n\,[u(n + 10) - u(n - 20)]$

$X2(n) = \cos(0.1\pi n)\,[u(n) - u(n - 30)]$

$X3(n) = (1.2)^n\,[u(n + 5) - u(n - 10)]$

Experiment NO.6

AIM: Auto-and Cross-correlation function

Response of LTI system from constant coefficient difference equation

Correlation:
The cross-correlation and auto correlation between two sequences *x (n)* and *y (n)* is given as:

$\gamma yx (l) = y (n) * x (-n)$

$\gamma xx (l) = x(n) * x(-n)$

Correlation can be computed using **conv _ m** function if sequences are finite duration.

Difference Equation:
An L TI discrete time system can also be described by a constant coefficient difference equation of the form:

$$\sum_{K=0}^{N} a_k\, y (n-k) = \sum_{m=0}^{M} b_m\, x (n-m) \text{ For ¥ n}$$

Where $a_o = 1$, and N= order of the difference equation.

This equation describes a recursive approach for computing the current output from the input values and previously computed output values.

MATLAB implementations:

A MATLAB built-in function called filter is available to solve difference equations numerically, where the input arguments are the input sequence *x (n),* difference equation coefficients band a. the function is invoked by

y=fiIter (b, a, x) % b= [b_o b_I b_2 ....b_M] a row-vector of size M+ 1
% a= [ao a] a2 ....AN] a row-vector of size N+l
% x= [... xo Xl X2 " ....] an input row-vector
% y= [....Yo Yl Y2 ....] an output row-vector of size x

For FIR filter: a=l

For IIR all-pole filter: b=l

The filter function only provides the zero-state response of the system.

Impulse response hen) of an LTI discrete system:

If the input to the LTI system is an impulse sequence i.e. *x (n) = 5(n),* the invoking filter function, it returns the values of the impulse response sequence.

h=fiIter (b, a, delta) % delta is an input row contains the impulse sequence

Exercise:

Q.l. Detem1ine the auto-correlation sequence $\gamma_{xx}(l)$ and the cross-correlation sequence $\gamma_{xy}(l)$ for the following sequences and plot them using **stem** function

$X(n) = (0.9)^n,\ 0 \le n \le 20;$

$Y(n) = (0.8)^{n};\ -20 \le n \le 0;$

Comment on your observation.

Q.2. A particular discrete LTI system is described by the difference equation

$$y(n) - 0.5y(n-1) + 0.25\,y(n-2) = x(n) + 2x(n-1) + x(n-2)$$

(a) Detem1ine and plot the impulse response of the system over $0 \le n \le 100$.

(b) Check the stability of the system using the BIBO stability criteria and using the impulse Response sequence. Comment on the two answers.

(c) If the input to the system is $x(n) = [5 + 3\cos(0.2\pi n) + 4\sin(0.6\pi n)]\, u(n)$.

Determine response $y(n)$ over $0 \le n \le 200$.

Experiment No.7

Fourier Series Use of **rem** function

Rem: Remainder after division.

REM(x, y) is x - n. *y where n = fix (x. /y) if y ~= 0. (y not equal to zero)
If y is not an integer and the quotient x. /y is within round off error of an integer, then n is that integer. By convention, REM(x,0) is NaN. The input x and y must be real arrays of the same size, or real scalars.

REM(x,y) has the same sign as x while MOD(x,y) has the same sign as y. *REM(x,y)* and MOD(x,y) are equal if x and y have the same sign, but differ by y if x and y have different signs.

Example: Generate a Pure Square Wave, with time period T=1 sec, sampling rate = 0.01 and Max time = 5 with maximum amplitude of 0.5 wave get first negative pulse from 0.25 to 0.5 , rest of time it is on positive cycle using **rem** tag of Matlab
MATLAB code

```
-------------------------------------------------------------------------------------------------------
t= (0:0.001:5);
x= (rem (t, 1)>=0.5 | rem (t, 1) <0.25)-0.5;
% (0.5=< rem (t, 1) OR rem (t, 1) < 0.25)
%if it true then it return a 1, otherwise a0.
plot (t, x);
title ('Pure Square Wave, sampling rate = 0.001 and Max time = 3');
xlabel ('time'), ylabel ('x ');
axis ([0, 5, -0.6, 0.6]);
-------------------------------------------------------------------------------------------
```

Q.1. Generate a Pure Square Wave, with time period T=2 sec, sampling rate = 0.01 and Max time = 10 with maximum amplitude of 0.75 and minimum amplitude -0.25. The first negative pulse must start from 0.5 to 0.75, and remain positive for the rest of cycle period. Generate the square wave using **rem** function

Synthesizing a periodic function using Fourier series

Q.l A symmetric square wave can be approximated by a sum of the first M harmonics:

$$x(t) = \sum_{K=1}^{M} C_k \cos(2\pi kt)$$

The Fourier series coefficients for these harmonics are given by:

$$C_k = \begin{cases} (2/k\pi)\,(-1)^{(k-1)/2}, & k \text{ odd} \\ 0, & k \text{ even} \end{cases}$$

Generate Square Wave by accumulating these Fourier coefficients.

Experiment No.8

Fourier series:

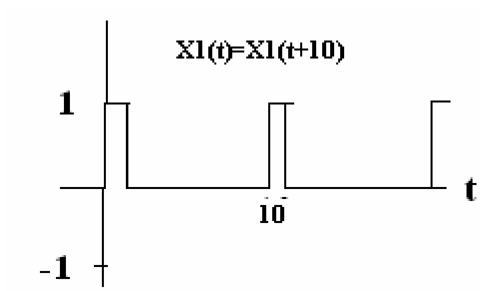
Synthesize the periodic function using Fourier series Coefficients:

Q.l.For given a_k's plots the signal x (t).

(a)

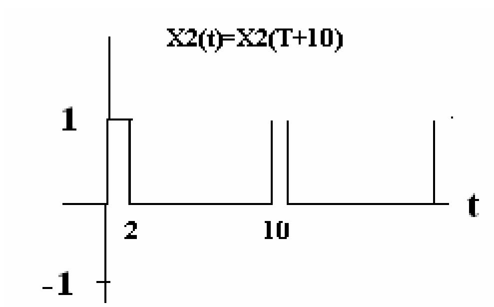
a_0 =1/10 exp (– jk π / 10)

a_K = {sin (k π/ 10) / kπ} exp (– jk π / 10)

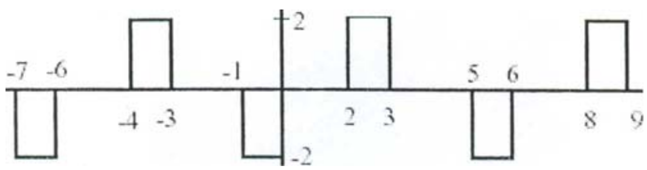


(b) a_0^2 =1/5 exp (-jkπ/5)

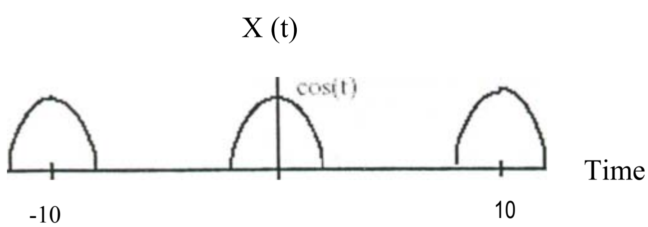
a_k^2 = {sin (k π/ 5) / kπ} exp – jk (π / 5)



Q2. ak = (- 4j/k*pi) sin (pi *k/6) sin (pi*k/2) exp(-j*k*pi/3)



Q3. ak = cos (pi/2*k*w0)/5/ (1-(k*w0) ^ 2)



Experiment NO.9

DTFT

Aim: ***MATLAB Implementation of discrete time Fourier transform (DTFT) of discrete time signals***

DTFT of infinite duration exponential signals

Let $x(n) = a^n u(n)$

If the sequence $x(n)$ is absolutely summeble (for a <1), its DTFT exists.

$$\text{Thus} \quad X(\omega) = \sum_{n=-\infty}^{n=\infty} x(n)\, e^{-j\omega n}$$

$$= 1/(1-ae^{-j\omega}) = e^{j\omega} / (e^{-j\omega} - a)$$

MATLAB Implementation

If x (n) is of infinite duration, then MATLAB cannot be used directly to compute $X(\omega)$ from x(n). However, we can use it to evaluate the expression $X(\omega)$ over [0, π] frequencies and then plot its magnitude and phase / or real and imaginary parts.

A MATLAB given below for computing DTFT of infinite duration sequence $x(n)$ = (0.6r $u(n)$ at the frequency points w=[O: 1 :200] *pi/20 1.

MATLAB script

```
M=100;
k=0: M-1;
w= (pi/M)*k;
X=exp (j*w). / (exp (j*w)-0.6*ones (1, M));
 w= (pi/M)*k;
Subplot (2, 1, 1)
Plot (w/pi, abs(X));
Xlabel (‘normalized frequency');
ylabel ('Amplitude');
Title ('Magnitude plot');
Subplot (2, 1, 2)
Plot (w /pi, angle(X)/pi);
Xlabel (‘normalized frequency');
ylabel ('phase');
Title ('phase plot');
```

DTFT of a Finite Duration Sequence:

If X (n) is of finite duration, then MATLAB can be used to complete X (w) numerically at any frequency w. We calculate X9w) at equispaced frequencies between ((0,π) and then X9w) can be computed as matrix-Vector operation.

Let $\omega_{K} = 2^{*} (\pi / M) k$ K= 0, 1, 2,……………….M-1

$$\text{Then } X(\omega) = \sum_{N=n1}^{n2} x(n)\, e^{-j (2\pi/ M) Kn}$$

N= n1: n2;

In *Malix-Vector* form

X= xW

Where x = [1 *N] row vector
X= [l *N] row-vector
W= [N*M] Matrix

MATLAB Implementations:

X=x*(exp (-j *2 *pi/M)). ^ (n'*w);

MATLAB Exercises:

Q.l. Write a MATLAB function to compute the DTFT of finite duration sequence. The format of the function should be

Use this function to compute the DTFT in the following problems wherever required.

Q.2. For each of the following sequences determine the DTFT of *x(n)*. Plot the magnitude and phase of *X (w)*.

(a) $X(n) = 2^{0.8n}\{u(n)-u(n-20)\}$ (b)

$x(n) = n^{0.9n}(u(n)-u(50)\}$

(c) $x(n)=\{4,3,2,1,2,3,4\}$. Comment on the angle plot.

(d) $x(n) = (4,3,2,1,0,-1,-2,-3,-4)$. Comment on the angle plot.

Q.3. Detennine DTFT of each of the following sequences and plot the magnitude and phase of *X(w)*.

(a) $X(n) = 3(0.9)^n u(n)$

(b) $X(n) = 5(-0.9)^n \cos(0.1\pi n)\, u(n)$.

Q4. A symmetric rectangular pulse is given by

$$RN(n) = \begin{cases} 1, & -N \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

Detem1ine the DTFT for $N=5, 15, 25, 100$. Scale the DTFT so that $X(O) = 1$. Plot the normalized DTFT over [-π,π[]. Study these plots and comment on their behavior as a function of N.

Q.5. Repeat question Q.3 for a symmetric triangular pulse that given by

$$T_N(n) = (1-|n|/N)\, R_N(n)$$

Q.6. Repeat question Q.3 for a symmetric raised cosine pulse that given by

$$C_N(n) = [0.5 + 0.5\cos\{n\pi/N)\; RN(n)$$

Experiment No.1 0

Aim: ***MATLAB Implementation of Discrete Fourier Transform of LTI systems***

Frequency Response

The DTFT of an impulse response is called Frequency Response

$$h(n) \overset{\text{DIFT}}{\longleftrightarrow} H(\omega)$$

Response to Sinusoidal sequences:

Let $x(n) = A\cos(\omega 0 n+\theta_o)$ be an input to an LTI system $h(n)$. Then the *response* $y(n)$ to the input $x(n)$ is another sinusoid of the same frequency ω_{0o} with amplitude gained by $|H(\omega 0)|$ and phase shifted by ∟$H(\omega 0)$, that is

$$Y(n) = A|H(\omega o)| \cos(\omega o n + \theta o + ∟H(\omega_o))$$

This response is called *steady state response,* which is denoted by $Yss(n)$.

If x(n) is a linear combination of sinusoids, then response yen) also be a linear combination of sinusoids. Thus,

$$\sum_k A_k \cos(\omega_k n + \theta_k) \longrightarrow \boxed{H(w)} \longrightarrow \sum_k A_k |H(\omega_k)| \cos(\omega_k n + \theta_k + \angle H(\omega_k))$$

Frequency Response Function from Difference Equation:

The input output of an LTI discrete system can be represented through difference equation

$$Y(n) = -\sum_{l=1}^{N} a1\, y(n-1) + \sum_{m=0}^{m} bm\, x(n-m)$$

Taking DTFT on both side we have

$$H(w) = \frac{\sum_{m=0}^{m} bm\, e^{-jwm}}{1 + \sum_{l=1}^{N} a1\, e^{-jwm}}$$

MA TLAB Implementation

If we evaluate *H (ω)* at equispaced frequencies over *[0, π]* or *[-π π,]* then *H (ω)* can easily implemented in MATLAB by Matrix-Vector multiplication.

Write a MATLAB function frqresp (b, a, w) to implement the above relation. A MATLAB script of the frqresp function is given below

---

```
Function [H] = freqresp (b, a, w)
% Frequency response function from difference equation
%--------------------------------------------------
% [H] =freqresp (b, a, w)
% H=frequency response array evaluated at w frequency locations


% b=numerator coefficient array
% a=denominator coefficient array (a (0) =l)
 % w=frequency location array M=length (w);
k=O: length (b)-1;
I=O: length (a)-I;
p=2*pi/M;
Num=b*(exp (-j*p)). ^ (k'*w);
den=a*(exp (-j*p)) ^ (l'*w);
H=num. /den;
```

MA TLAB Exercises:

Q.l. For each of the LTI systems described by the impulse response, determine the Frequency response function $H(\omega)$ and plot the magnitude response $|H(\omega)|$ and phase response $\angle H(\omega)$.

(a) $h(n) = (0.9)^{|n|}$

(b) $h(n0) = \sin c\,(0.2n)\,[u(n+20) - u(n-20)]$

(c) $h(n) = [(0.5)^{n} + (0.4)^{n}]u(n)$

(d) $h(n) = (0.5)^{n1} \cos(0.1n\pi)$

Q.2. Let $x(n) = 3\cos(0.5\pi n + 60°) + 2\sin(0.3\pi n)$ be the input to each of the systems described in question Q.1. In each case determine the output $y(n)$.

Q.3. Determine $H(\omega)$ and plot its magnitude and phase for each of the following systems.

(a) $Y(n) = \sum_{m=0}^{6} x(n-m)$

(b) $Y(n) = x(n) + 2x(n-1) + x(n-2) - 0.5y(n-1) - 0.25y(n-2)$

(c) $Y(n) = x(n) - \sum_{l=1}^{5} (0.5)^{l} y(n-l)$

QA. A LTI system is described by the difference equation

$$Y(n) = -\sum_{l=1}^{3} (0.81)^{l}\, y(n-2l) + \sum_{m=0}^{3} b_m\, x(n-2m)$$

Determine the steady state response of the system to the following inputs:

(a) $x(n) = 5+10(-1)^n$

(b) $X(n) = 1 + \cos(0.5\pi n + \pi/2)$

(c) $X(n) = 2\sin(\pi n/4) + 3\cos(3\pi n/4)$

(d) $x(n) = \cos(\pi n)$

In each case generate $x(n)$, $0 \le n \le 200$ and process it through the filter function to obtain $y(n)$. Compare your $y(n)$ with steady-state response in each case.