LAB # 5 HANDOUT

1. FAST FOURIER TRANSFORM

Data Sequence \( A_0 = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\} \)

\[ W_N = e^{-\frac{2\pi i}{N}} \]
\[ W_{N/2}^2 = W_N \]
\[ W_N^{(k+N/2)} = -W_N^k \]

Then, the following formulas must be used.

\[ X_1(k) = X_{11}(k) + W_N^k X_{12}(k) \quad k=0, \ldots, 7 \]
\[ X_{11}(k) = X_{21}(k) + W_{N/2}^k X_{22}(k) \quad k=0,1,2,3 \]
\[ X_{12}(k) = X_{23}(k) + W_{N/2}^k X_{24}(k) \quad k=0,1 \]

\[ X_{21}(k) = x_0 + W_{N/4}^k x_4 \]
\[ X_{21}(0) = x_0 + x_4 \]
\[ X_{21}(1) = x_0 - x_4 \]
\[ X_{22}(k) = x_2 + W_{N/4}^k x_6 \]
\[ X_{22}(0) = x_2 + x_6 \]
\[ X_{22}(1) = x_2 - x_6 \]

By using "fft" and "ifft" functions in matlab:

\[ x = [1\ 1\ 0\ 0\ 0\ 0\ 1\ 1] \]
\[ >> \text{fft}(x) \]
\[ ans = \]
\[ Columns 1 through 8 \]
\[ 4.0000 \quad 2.4142+1.0000i \quad 0 \quad -0.4142-1.0000i \quad 0 \quad -0.4142+1.0000i \quad 0 \quad 2.4142-1.0000i \]
\[ >> \text{ifft}(ans) \]
\[ ans = \]
\[ 1.0000 \quad 1.0000 \quad 0 \quad 0 \quad 0.0000 \quad 0 \quad 1.0000 \quad 1.0000 \]
2. THE BUTTERFLY

The butterfly that shows all of the connections is shown below:

\[ X_{21}(0) \]
\[ X_{22}(0) \]
\[ X_{21}(1) \]
\[ X_{22}(1) \]

---

\[ X_{11}(0) = X_{21}(0) + W_8^0 X_{22}(0) \]
\[ X_{11}(2) = X_{21}(0) - W_8^0 X_{22}(0) \]
\[ X_{11}(1) = X_{21}(1) + W_8^2 X_{22}(1) \]
\[ X_{11}(3) = X_{21}(1) - W_8^2 X_{22}(1) \]
3. SAMPLING OF A SINUSOIDAL SIGNAL

In this program we will investigate the sampling of a continuous-time sinusoidal signal \( x_a(t) \) at various sampling rate. Since MATLAB cannot strictly generate a continuous-time signal, we will generate a sequence \( \{x_a(nT_H)\} \) from \( x_a(t) \) by sampling it at a very high rate \( T_H \) such that the samples are very close to each other. A plot of \( x_a(nT_H) \) using the plot command will then look like a continuous-time signal.

As a result, \( x[n] \) is generated by periodically sampling \( X_a(t) \) at uniform time interval \( T \).

\[
x[n] = X_a(nT)
\]

\( T = \) sampling interval  
\( F_s = \) sampling rate, sampling frequency = \( 1/T \)

**M-file:**

```matlab
% Illustration of the Sampling Process in the Time-Domain
clf; % Clear current figure
clear all; % Clear all variables
%--------------------------------------------------------
% Generation of the continuous-time signal
%--------------------------------------------------------
t = 0 : 0.0005 : 1;
f = 13;
xa = cos(2*pi*f*t);
subplot(2,1,1)
plot(t,xa);grid
xlabel('Time, msec');
ylabel('Amplitude');
title('Continuous-time signal x_a(t)');
axis([0 1 -1.2 1.2]);
subplot(2,1,2)
%--------------------------------------------------------
% Sampling in time-domain
%--------------------------------------------------------
T = 0.1;
n = 0 : T : 1;
xs = cos(2*pi*f*n);
k = 0 : length(n)-1;
estem(k,xs,'r');grid
xlabel('Time index n');
ylabel('Amplitude');
title('Discrete-time signal x[n]');
axis([0 length(n)-1 -1.2 1.2])
```
4. ALIASING EFFECT IN THE TIME-DOMAIN

In this program we will generate a continuous-time equivalent $y_a(t)$ of the discrete-time signal $x[n]$ generated in previous program to investigate the relation between the frequency of the sinusoidal signal $x_a(t)$ and sampling period. To generate the reconstructed signal $y_a(t)$ from $x[n]$, we pass $x[n]$ through an ideal lowpass filter. The higher frequency analog signal can be reconstructed in lower freq. form after sampling.

**Formula of Reconstruction:**

$$X_a(t) = \sum_{n=\infty}^{n=\infty} x(n) \text{sinc}[F_s(t - nT_s)]$$

To prevent aliasing $f_s > 2f_0$.
% Illustrating of Aliasing Effect in the Time-Domain

clf;
clear all;
T = 0.1;
f = 13;
n = (0:T:1)';
xs = cos(2*pi*f*n);

%------------------------------------------------------
% linspace(x1,x2,N) generates N points between x1 and x2.
% for N<2, linspace returns x2.
%------------------------------------------------------
t = linspace(-0.5,1.5,500)';

%------------------------------------------------------
% reconstruction formula is shown below.
%------------------------------------------------------
ya = sinc((1/T)*t(:,ones(size(n))) - (1/T)*n(:,ones(size(t))))'*xs;
plot(n,xs,'o',t,ya);grid;
xlabel('Time index n');
ylabel('Amplitude');
title('Reconstructed continuous-time signal \( y_a(t) \)');
axis([0 1 -1.2 1.2])
5. EFFECT OF SAMPLING IN THE FREQUENCY-DOMAIN

In order to convert a continuous-time signal $x_a(t)$ into an equivalent discrete-time signal $x[n]$, the former must be band-limited in the frequency-domain. To illustrate the effect of sampling in the frequency-domain we choose an exponentially decaying continuous-time signal with a CTFT that is approximately band-limited.

**M-file:**

```matlab
% Illustrating of the Aliasing Effect
% In the Frequency-Domain
clf;
clear all;
%--------------------------------------------------------
% Generation of the continuous-time signal
%--------------------------------------------------------
t = 0 : 0.005 : 10;
xa = 2*t.*exp(-t);
subplot(2,2,1)
plot(t,xa);grid    % grid is used for grid lines
xlabel('Time, msec');
ylabel('Amplitude');
title('Continuous-time signal $x_{a}(t)$');
subplot(2,2,2)
wa = 0 : 10/511 : 10;
%--------------------------------------------------------
% H=freqs(B,A,W) returns the complex frequency response
% vector H of the filter B/A; the frequency response is
% evaluated at the points specified in vector W.
%--------------------------------------------------------
Ha = freqs(2,[1 2 1],wa);
plot(wa/(2*pi),abs(Ha));grid
xlabel('Frequency, KHz');
ylabel('Amplitude');
title('|$X_{a}(j\Omega)$|');
axis([0 5/pi 0 2]);
subplot(2,2,3)
%--------------------------------------------------------
% Sampling of the signal
%--------------------------------------------------------
T = 1;
n = 0 : T : 10;
xs = 2*n.*exp(-n);
k = 0 : length(n)-1;
stem(k,xs);grid;
xlabel('Time index n');
ylabel('Amplitude');
title('Discrete-time signal x[n]');
subplot(2,2,4)
wd = 0 : pi/255 ; pi;
```
Hd = freqz(xs,1,wd);
plot(wd/(T*pi),T*abs(Hd));grid;
xlabel('Frequency, KHz'); ylabel('Amplitude');
title('|X(e^{j\Omega})|');
axis([0 1/T 0 2]);