Experiment 2
Sampling Theorem in Frequency Domain

Aim

To study the sampling theorem in frequency domain and aliasing effects by sampling the spectrum of DT signal.

Theory

Consider a discrete time (DT) signal $x(n)$ with L number of samples. Discrete time Fourier transform (DTFT) of $x(n)$ will be periodic with period equal to 2π. One period of DTFT is sampled using N number of samples (DFT). If $N > L$, then it is possible to recover signal $x(n)$ from the DFT samples.

If the sampling theorem is not obeyed, that is, if $N < L$, then there is aliasing in the time domain. The replicated time domain signal will overlap and will generate aliasing effect.

Let us consider a known sequence $x(n) = a^n u(n)$ with $a = 0.8$. We will calculate $X(\omega)$ and sample it at N equally spaced points. Let $N = 15$. Clearly $N = 15$ is less than the number of samples in $x(n)$, that is, ∞. So we will be in a position to observe aliasing.

For detailed theory refer to Example 1 in Chapter 5, Section 5.1.

Experiment

Write a MATLAB program to plot the signal $x(n) = a^n u(n)$. We will plot only first 15 samples. We have to write a program in MATLAB to plot the signal. This is illustrated in the sample program here. We are generating some 15 samples of the signal and plotting it for the value of $a = 0.8$.

```
    a = 0.8;
for n = 1:15,
  x(n) = (a)^n;
end
stem(x);title('plot of x(n) = (a)^n');xlabel('sample no');ylabel
('amplitude');
```

The plot will be as shown in Figure 1. The signal actually has infinite points. When we take 15-point DFT, we are using $N < \infty$. There will be aliasing. Now plot a 15-point DFT using MATLAB program. It is as shown in Figure 2.

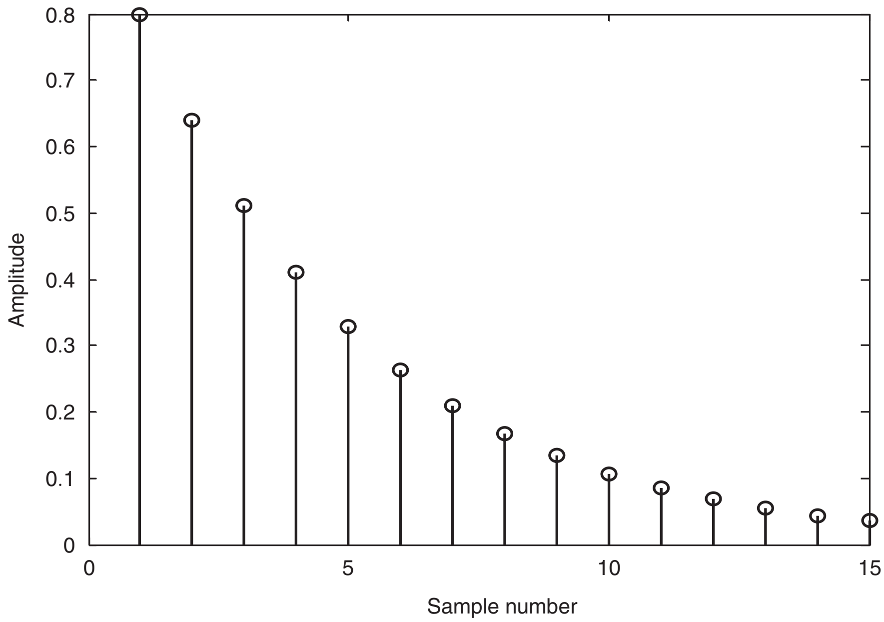

Figure 1 Plot of 15 signal samples.

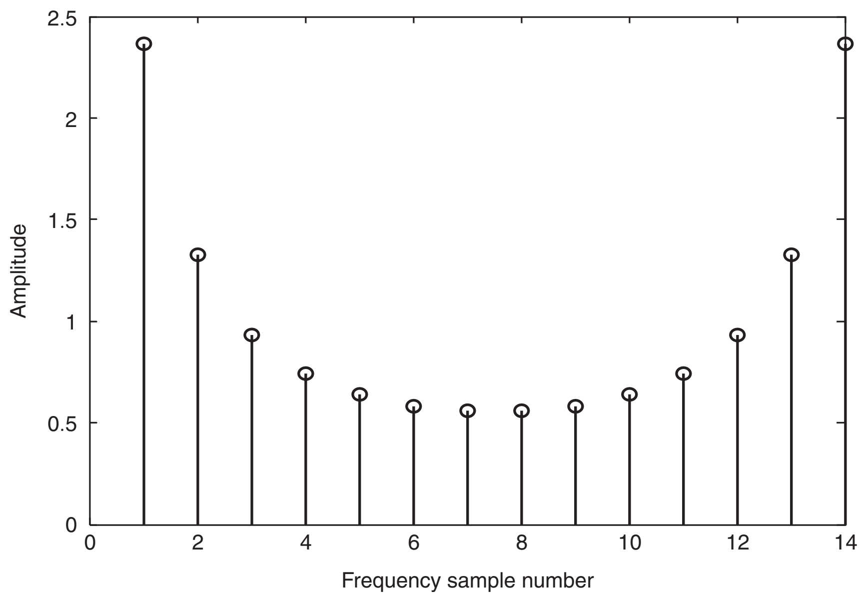

Figure 2 Plot of 15-point DFT.

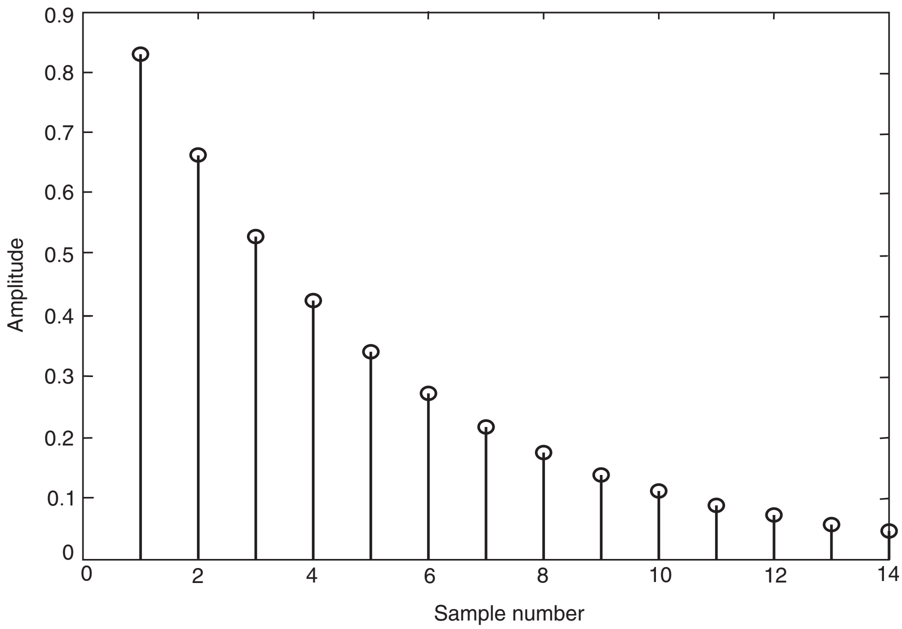

Figure 3 Recovered signal samples.

Teaser *The reader is encouraged to write a program to plot DFT output using "fft" command in MATLAB.*

We have sampled the DTFT (DFT) using number of points $N < \infty$. When we revert back in time domain using the following equation, the recovered samples will be aliased as shown in Figure 3.

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN) = \sum_{l=-\infty}^{0} a^{n-lN} = a^n \sum_{l=0}^{\infty} a^{lN} = \frac{a^n}{1-a^N} \tag{1}$$

We have to write a MATLAB program to calculate 15 samples of $x(n)$ using Eq. (1). Table 1 shows the original signal samples of $x(n)$ and reconstructed sample values using Eq. (1).

Table 1 First three samples of original signal and reconstructed signal

Sample Number	*1*	*2*	*3*	*4*
Original signal	0.8000	0.6400	0.5120	–
Recovered signal	0.8292	0.6633	0.5307	–

We have calculated first three samples only. The signal exists for further values of n, say up to infinity. The repeated signal $x_P(n)$ shows 15 sample values repeated. Actually the repeated signal overlaps onto itself and generates aliasing effect.

For plot of $x_P(n)$ and its analysis refer to Example 1 in Chapter 5, Section 5.1

We see that the reconstructed signal samples have higher values as they are the aliased samples. The value of reconstructed sample at $n = 1$ is obtained as

$$\begin{aligned} x(1)+x(16)+\ldots &= 0.8+(0.8)^{16}+(0.8)^{32}+(0.8)^{64} \\ &= 0.8+0.0281+0.00079+\ldots \\ &= 0.8292 \end{aligned}$$

Similarly, we can calculate value of second and third samples also. The second sample for $n = 2$ is given as

$$\begin{aligned} x(2)+x(17)+\ldots. &= (0.8)^{2}+(0.8)^{17}+(0.8)^{33}+(0.8)^{65}+\ldots. \\ &= 0.64+0.0225+0.000634+\ldots \\ &= 0.6633 \end{aligned}$$

Teaser *The reader is encouraged to find the value of a third sample on similar lines and confirm that it is equal to 0.5307 and so on for further samples.*

The MATLAB program for aliasing in time domain using sampling theorem in frequency domain is as follows.

```
%alasing in time domain-sampling theorem in frequency domain
clear all;
N=15;
for i=1:15,
    x(i)=(0.8)^i;
end
stem(x);
xlabel('sample no.');
ylabel('amplitude');
title('plot of a^n');
figure;
for i=1:14,
    y(i)=abs(1/(1-(0.8)*(exp(-j*2*pi*i/N))));
end
stem(y);
xlabel('freq. sample no.');
ylabel('amplitude');
title('plot of DFT coefficients');
for i=1:14,
x1(i)=(0.8)^i/(1-(0.8)^15);
```

```
end
figure;
stem(x1);
xlabel('sample no.');
ylabel('amplitude');
title('plot of reconstructed x(n)');
for i=1:3,
disp(x(i));
disp(x1(i));
end
x2=x(1)+(0.8)^14;
disp(x2);
sum=0.0;
for w=0.1:0.1:2*pi,
    for l=1:15,
        sum(w)=1/(1-((0.8)*exp(-j*w)));
        end
end
figure;
plot(sum);
```