

clear all;

close all;

clc;

widthOfTheLine = 1.5;

numberOfWaves = 5;

messageSignalFrequency = input("Enter the Message Signal Frequency: ");

initialSamplingFreq = 50 * messageSignalFrequency;

timePerSample = 1 / initialSamplingFreq;

stopTime = 1;

timeAxis = 0:timePerSample:stopTime - timePerSample;

totalNumberOfSamples = size(timeAxis, 2);

samplingFrequencyInterval = initialSamplingFreq /
totalNumberOfSamples;

frequencyAxis = -initialSamplingFreq /

2:samplingFrequencyInterval:initialSamplingFreq / 2 -
samplingFrequencyInterval;

phiDegrees = 90;

phi = phiDegrees * pi / 180;

```
subplot(421);  
xData = sin(2 * pi * messageSignalFrequency * timeAxis + phi);  
plot(1000 * timeAxis, xData, "lineWidth", widthOfTheLine);  
set(get(gca, 'XLabel'), 'String', 'Time in milliSeconds (ms)');  
set(get(gca, 'YLabel'), 'String', 'Amplitude');  
set(get(gca, 'Title'), 'String', 'Un-Sampled Signal');  
grid on;  
axis([0 1000 * numberOfWaves / messageSignalFrequency -1 1]);
```

```
subplot(422);  
xDataFFT = fftshift(fft(xData));  
semilogx(frequencyAxis, abs(xDataFFT) / totalNumberOfSamples,  
"lineWidth", widthOfTheLine);  
set(get(gca, 'XLabel'), 'String', 'Frequency in Hertz (Hz)');  
set(get(gca, 'Title'), 'String', 'Un-Sampled Signal');  
grid on;  
axis([0 messageSignalFrequency + messageSignalFrequency / 3 0  
inf]);
```

```
underSamplingFrequency = 1.2 * messageSignalFrequency;
```

```

under_timePerSample = 1 / underSamplingFrequency;
under_timeAxis = 0:under_timePerSample:stopTime -
under_timePerSample;
under_totalNumberOfSamples = size(under_timeAxis, 2);
under_samplingFrequencyInterval = underSamplingFrequency /
under_totalNumberOfSamples;
under_frequencyAxis = -underSamplingFrequency /
2:under_samplingFrequencyInterval:underSamplingFrequency / 2 -
under_samplingFrequencyInterval;

```

```

subplot(423);
under_xData = sin(2 * pi * messageSignalFrequency *
under_timeAxis + phi);
hold on;
plot(1000 * under_timeAxis, under_xData, "lineWidth",
widthOfTheLine);
stem(1000 * under_timeAxis, under_xData);
set(get(gca, 'XLabel'), 'String', 'Time in milliSeconds (ms)');
set(get(gca, 'YLabel'), 'String', 'Amplitude');
set(get(gca, 'Title'), 'String', 'Under-Sampled Signal');

```

hold off;

grid on;

axis([0 1000 * numberOfWaves / messageSignalFrequency -1 1]);

subplot(424);

under_xDataFFT = fftshift(fft(under_xData));

semilogx(under_frequencyAxis, abs(under_xDataFFT) /
under_totalNumberOfSamples, "lineWidth", widthOfTheLine);

set(get(gca, 'XLabel'), 'String', 'Frequency in Hertz (Hz)');

set(get(gca, 'Title'), 'String', 'Under-Sampled Signal');

grid on;

axis([0 messageSignalFrequency + messageSignalFrequency /2 0
inf]);

perfectSamplingFrequency = 2 * messageSignalFrequency;

perfect_timePerSample = 1 /perfectSamplingFrequency;

perfect_timeAxis = 0:perfect_timePerSample:stopTime;

perfect_totalNumberOfSamples = size(perfect_timeAxis, 2);

perfect_samplingFrequencyInterval = perfectSamplingFrequency /
perfect_totalNumberOfSamples;

```
perfect_frequencyAxis = -perfectSamplingFrequency /  
2:perfect_samplingFrequencyInterval:perfectSamplingFrequency /2 -  
perfect_samplingFrequencyInterval;
```

```
subplot(425);
```

```
perfect_xData = sin(2 * pi * messageSignalFrequency *  
perfect_timeAxis + phi);
```

```
hold on;
```

```
plot(1000 * perfect_timeAxis, perfect_xData, "lineWidth",  
widthOfTheLine);
```

```
stem(1000 * perfect_timeAxis, perfect_xData);
```

```
set(get(gca, 'XLabel'), 'String', 'Time in milliseconds (ms)');
```

```
set(get(gca, 'YLabel'), 'String', 'Amplitude');
```

```
set(get(gca, 'Title'), 'String', 'Perfectly-Sampled Signal');
```

```
hold off;
```

```
grid on;
```

```
axis([0 1000 * numberOfWaves /messageSignalFrequency -1 1]);
```

```
subplot(426);
```

```
perfect_xDataFFT = fftshift(fft(perfect_xData));
```



```
semilogx(perfect_frequencyAxis, abs(perfect_xDataFFT) /  
perfect_totalNumberOfSamples, "lineWidth", widthOfTheLine);  
set(get(gca, 'XLabel'), 'String', 'Frequency in Hertz (Hz)');  
set(get(gca, 'Title'), 'String', 'Perfectly-Sampled Signal');  
grid on;  
axis([0 messageSignalFrequency + messageSignalFrequency / 2 0  
inf]);
```

```
overSamplingFrequency = 8 * messageSignalFrequency;  
over_timePerSample = 1 / overSamplingFrequency;  
over_timeAxis = 0:over_timePerSample:stopTime -  
over_timePerSample;  
over_totalNumberOfSamples = size(over_timeAxis, 2);  
over_samplingFrequencyInterval = overSamplingFrequency /  
over_totalNumberOfSamples;  
over_frequencyAxis = -overSamplingFrequency /  
2:over_samplingFrequencyInterval:overSamplingFrequency / 2 -  
over_samplingFrequencyInterval;
```

```
subplot(427);
```

```

over_xData = sin(2 * pi * messageSignalFrequency * over_timeAxis
+ phi);
hold on;
plot(1000 * over_timeAxis, over_xData, "lineWidth",
widthOfTheLine);
stem(1000 * over_timeAxis, over_xData);
set(get(gca, 'XLabel'), 'String', 'Time in milliseconds (ms)');
set(get(gca, 'YLabel'), 'String', 'Amplitude');
set(get(gca, 'Title'), 'String', 'Over-Sampled Signal');
hold off;
grid on;
axis([0 1000 * numberOfWaves / messageSignalFrequency -1 1]);

subplot(428);
over_xDataFFT = fftshift(fft(over_xData));
semilogx(over_frequencyAxis, abs(over_xDataFFT) /
over_totalNumberOfSamples, "lineWidth", widthOfTheLine);
set(get(gca, 'XLabel'), 'String', 'Frequency in Hertz (Hz)');
set(get(gca, 'Title'), 'String', 'Over-Sampled Signal');
grid on;

```