ELP311
Communication Engineering Laboratory

Experiment 8
Sampling using MATLAB

Ishvik Kumar Singh
2018EE10616

Objective

The aim is to understand basic sampling ideas, concepts related to interpolation and decimation.

MATLAB code

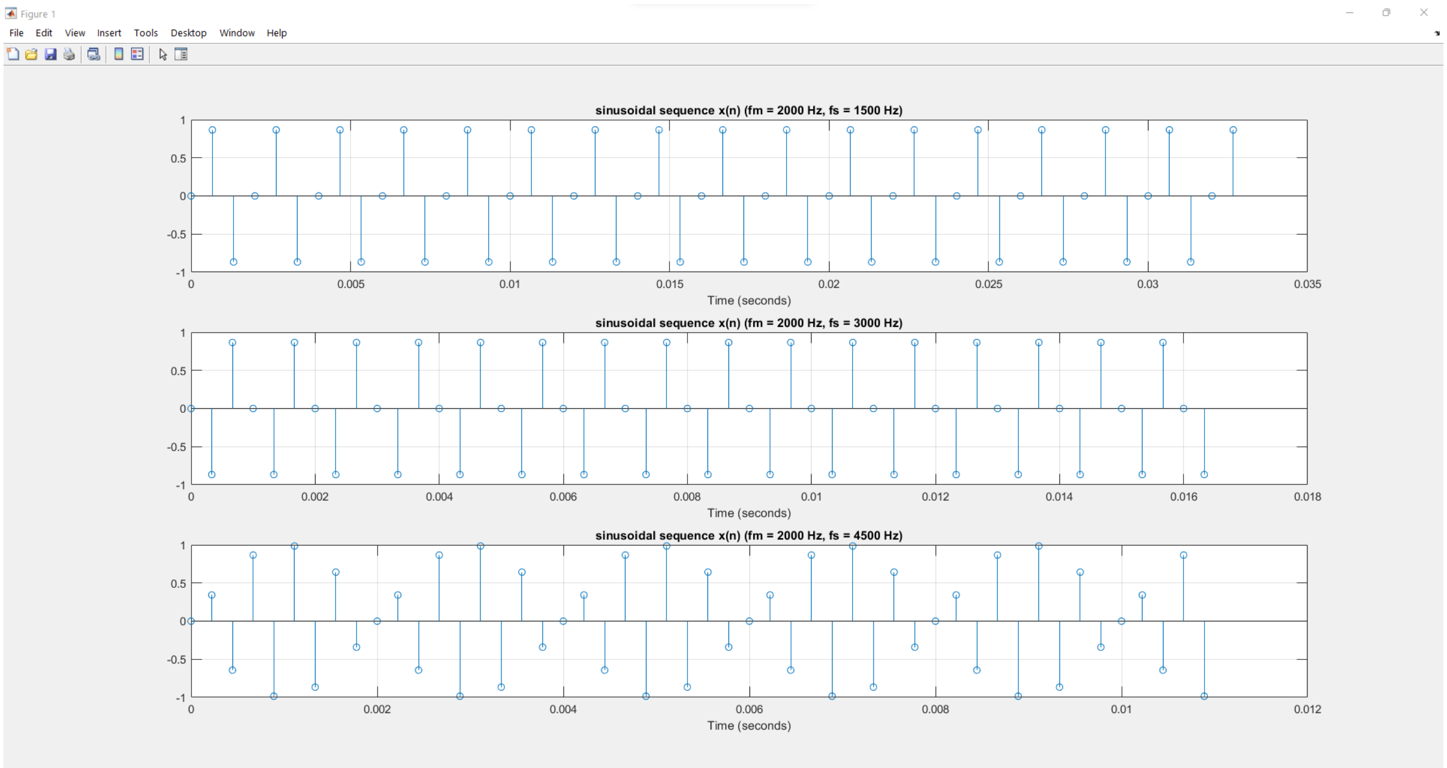
```
close all

fm  = 2000;
fs = [1500, 3000, 4500];

for i = (1 : 3)
    t1 = (0: 1/fs(i) : 0.5);
    x = sin(2* pi * fm * t1);

    figure (1);
    subplot(3, 1, i);
    stem(t1(1:50), x(1:50));
    title(sprintf('sinusoidal sequence x(n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    figure (2);
    subplot(3, 1, i);
    pspectrum(x, fs(i));
    title(sprintf('power spectrum of x(n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    grid;

    sound(x, fs(i));

    t2 = t1(2: 2: end);
    y = x(2: 2: end);

    figure (3);
    subplot(3, 1, i);
    stem(t2(1:50), y(1:50));
    title(sprintf('y(n) = x(2n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    figure (4);
    subplot(3, 1, i);
    pspectrum(y, fs(i));
    title(sprintf('power spectrum of y(n) = x(2n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    grid;

    sound(y, fs(i));

    z1 = kron(x, [1, zeros(1, 1)]);
```

```
    z2 = kron(x, [1, zeros(1, 2)]);

    t3_1 = 0: 1/(2 * fs(i)): 0.5;
    t3_2 = 0: 1/(3 * fs(i)): 0.5;

    figure (5);
    subplot(3, 2, 2 * i - 1);
    stem(t3_1(1:50), z1(1:50));
    title(sprintf('z(n) (fm = 2000 Hz, fs = %d Hz, L = 2)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    subplot(3, 2, 2 * i);
    stem(t3_2(1:50), z2(1:50));
    title(sprintf('z(n) (fm = 2000 Hz, fs = %d Hz, L = 3)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    figure (6);
    subplot(3, 2, 2 * i - 1);
    pspectrum(z1, fs(i));
    title(sprintf('power spectrum of z(n) (fm = 2000 Hz, fs = %d Hz, L = 2)', fs(i)));
    grid;

    subplot(3, 2, 2 * i);
    pspectrum(z2, fs(i));
    title(sprintf('power spectrum of z(n) (fm = 2000 Hz, fs = %d Hz, L = 3)', fs(i)));
    grid;

    sound(z1, fs(i));
    sound(z2, fs(i));

    sinc1 = (fs(i)/2) * sinc(2 * (fs(i)/4) * t3_1);
    sinc2 = (fs(i)/3) * sinc(2 * (fs(i)/6) * t3_2);

    f1 = conv(z1, sinc1);
    f2 = conv(z2, sinc2);

    z1_f = f1(1: length(z1));
    z2_f = f2(1: length(z2));

    sound(z1_f, fs(i))
    sound(z2_f, fs(i))

    figure (7);
    subplot(3, 2, 2 * i - 1);
    stem(t3_1(1:50), z1_f(1:50));
    title(sprintf('z(n) after filtering (fm = 2000 Hz, fs = %d Hz, L = 2)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    subplot(3, 2, 2 * i);
    stem(t3_2(1:50), z2_f(1:50));
```

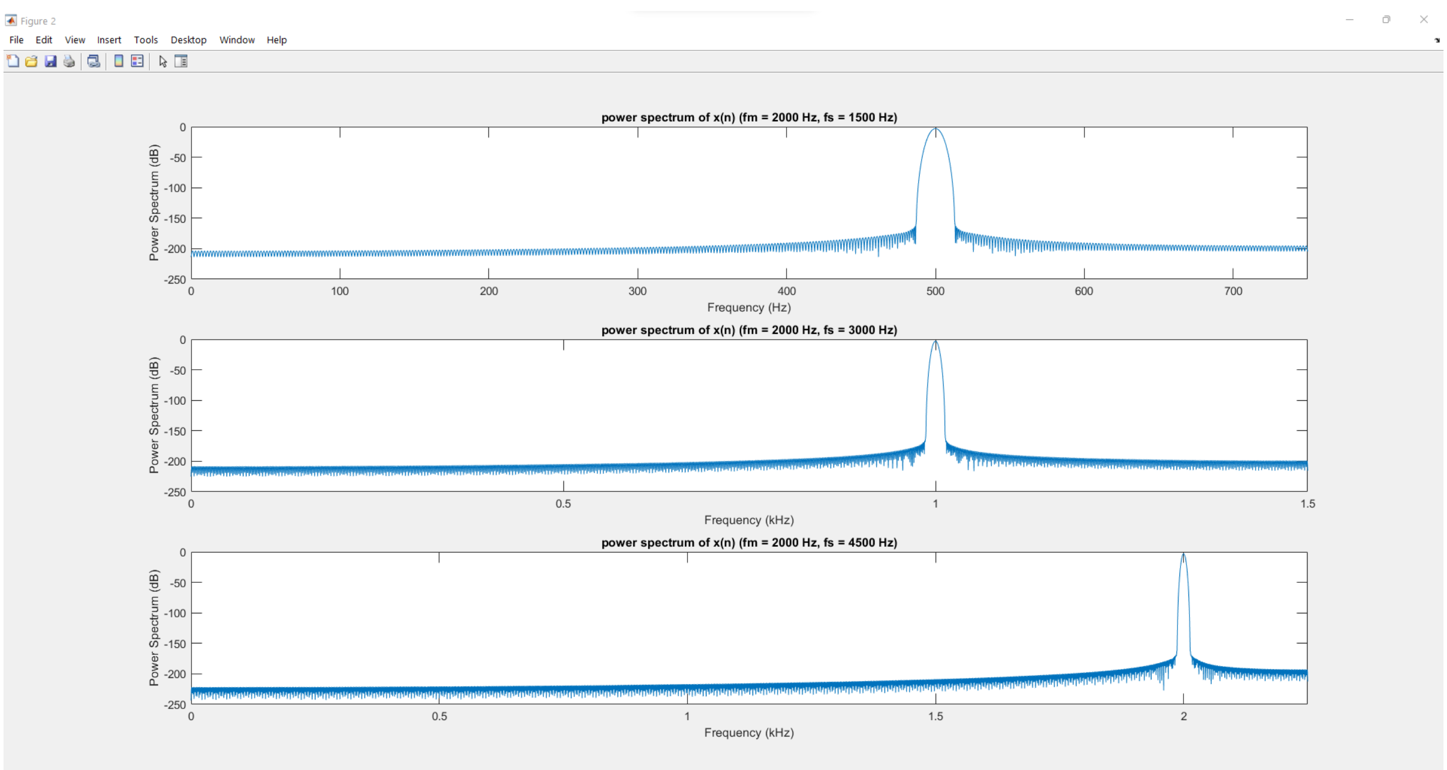
```
    title(sprintf('z(n) after filtering (fm = 2000 Hz, fs = %d Hz, L = 3)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    figure (8);
    subplot(3, 2, 2 * i - 1);
    pspectrum(z1_f, fs(i));
    title(sprintf('power spectrum of z(n) after filtering (fm = 2000 Hz, fs = %d Hz, L = 2)',
fs(i)));
    grid;

    subplot(3, 2, 2 * i);
    pspectrum(z2_f, fs(i));
    title(sprintf('power spectrum of z(n) after filtering (fm = 2000 Hz, fs = %d Hz, L = 3)',
fs(i)));
    grid;

    t4 = t3_2(2: 2: end);
    r = z2_f(2: 2: end);

    figure (9);
    subplot(3, 1, i);
    stem(t4(1:50), r(1:50));
    title(sprintf('r(n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    xlabel('Time (seconds)');
    grid;

    figure (10);
    subplot(3, 1, i);
    pspectrum(r, fs(i));
    title(sprintf('power spectrum of r(n) (fm = 2000 Hz, fs = %d Hz)', fs(i)));
    grid;

    sound(r, fs(i))
end
```

Plots

```
Part 1:
```

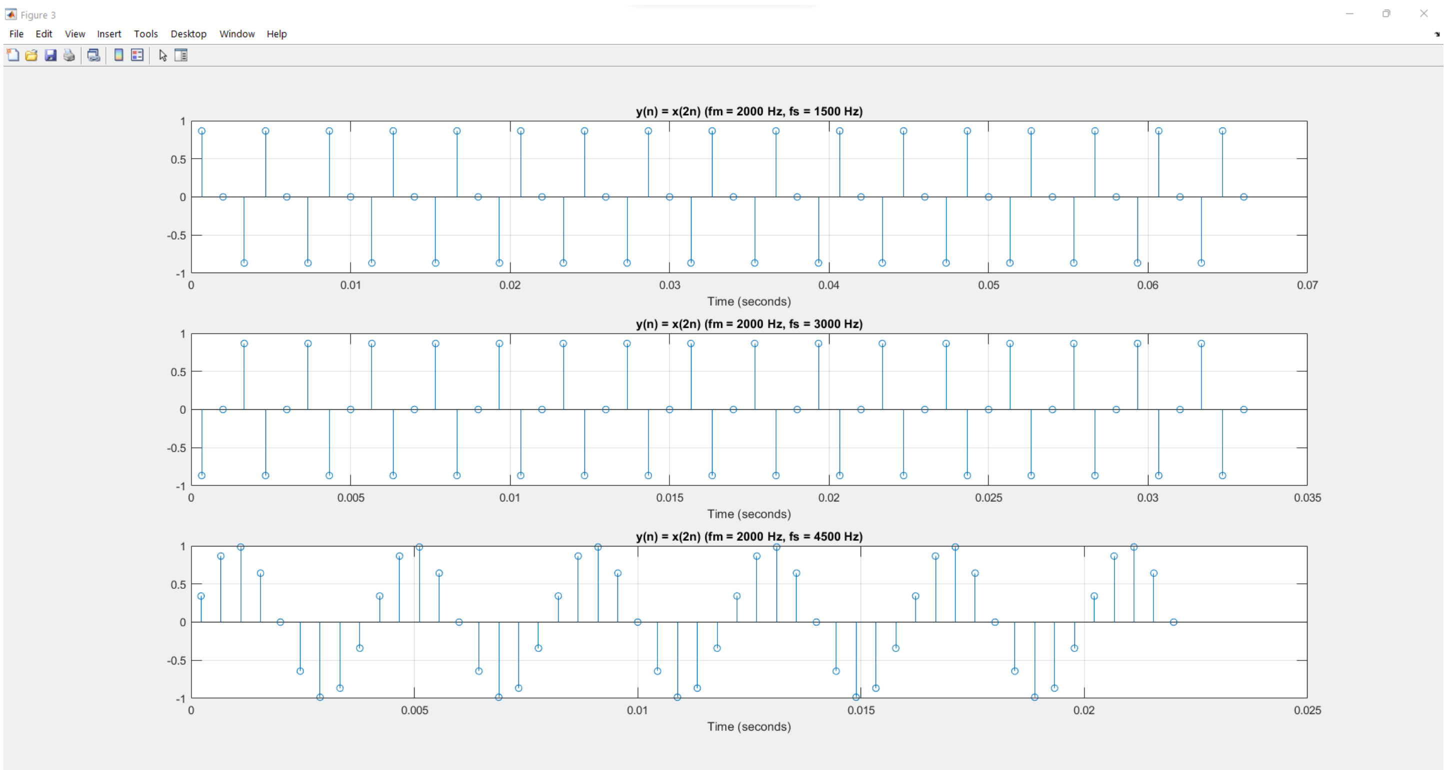


Plot of $x(n)$ at different sampling frequencies

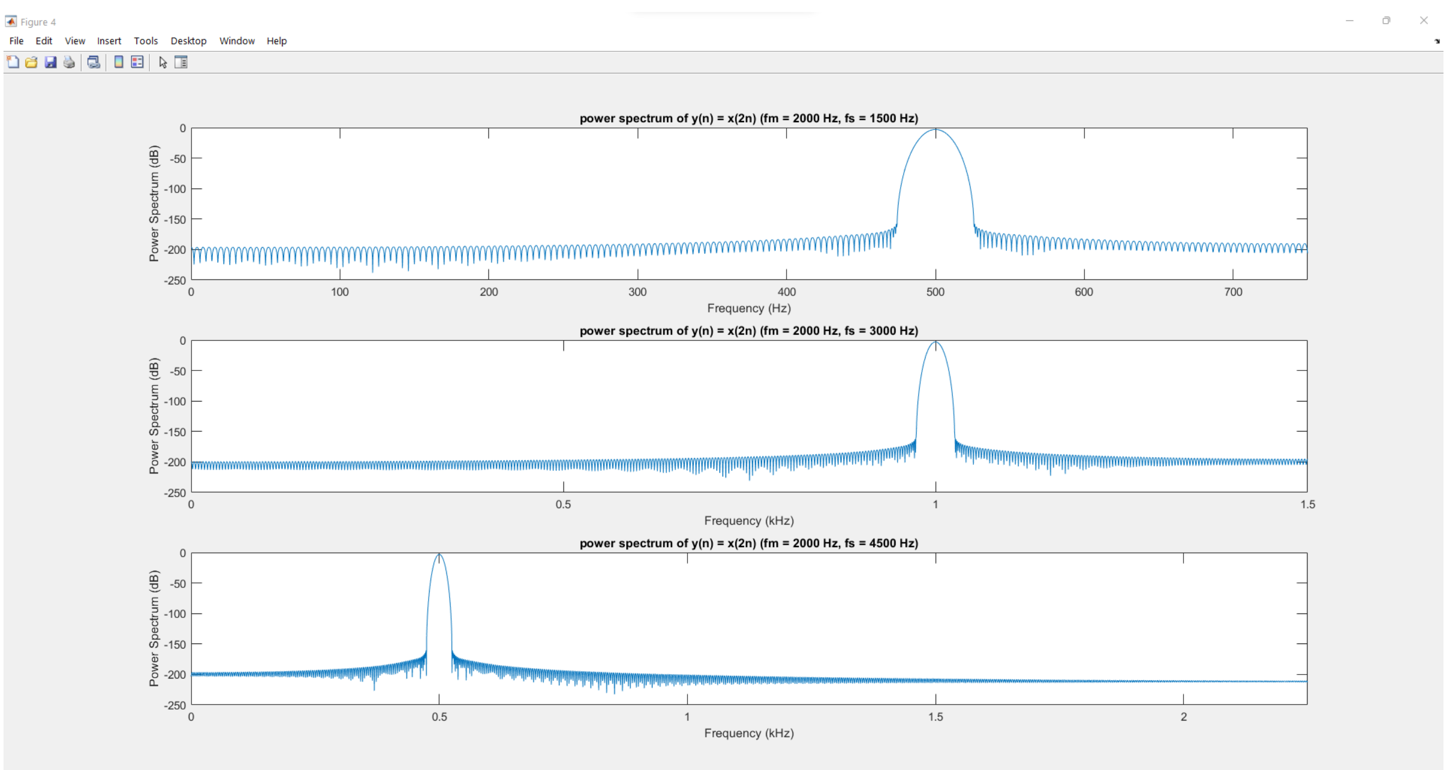


Power Spectrum of $x(n)$ at different sampling frequencies (aliasing can be observed for $f_s = 1500\text{ Hz}, 3000\text{ Hz}$)

Part 2:

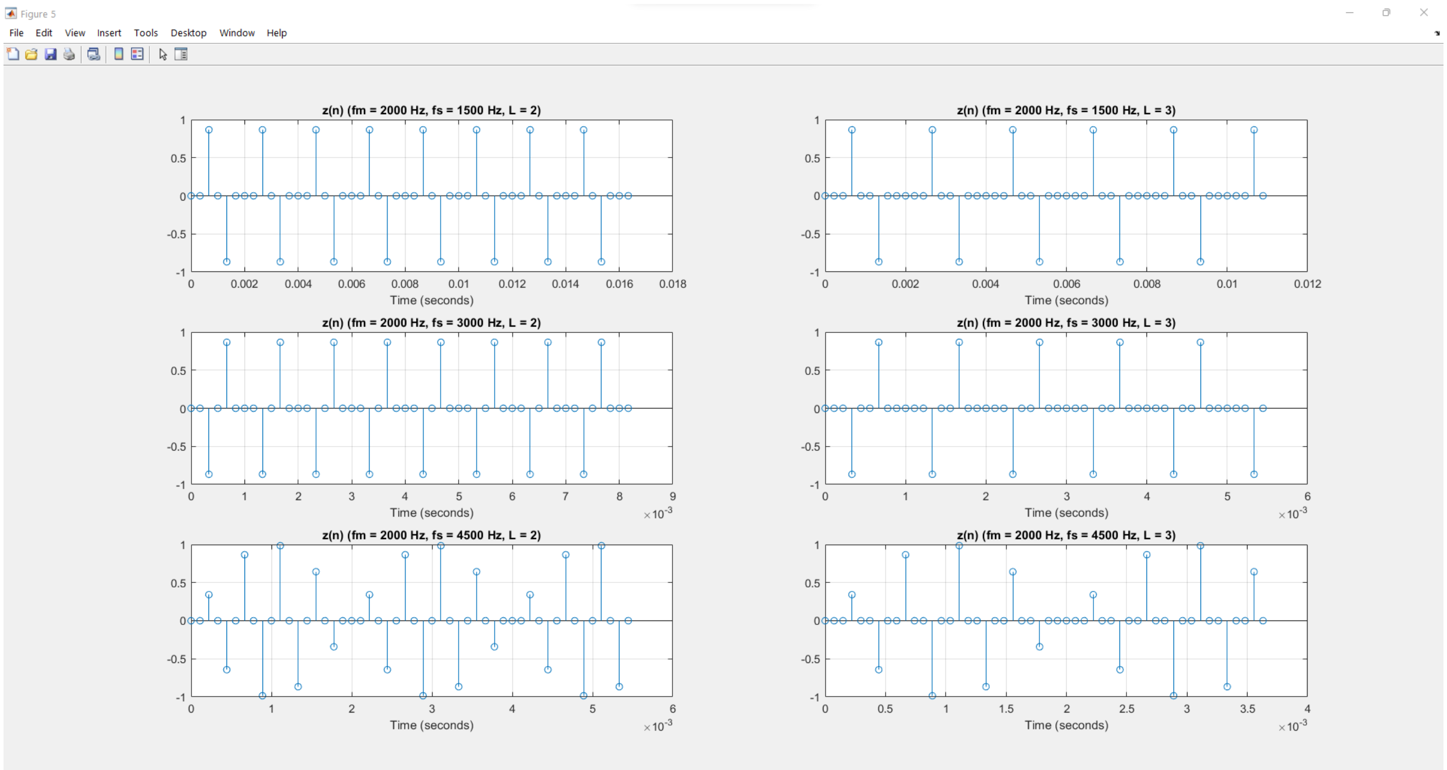


Plot of $y(n)$ at different sampling frequencies

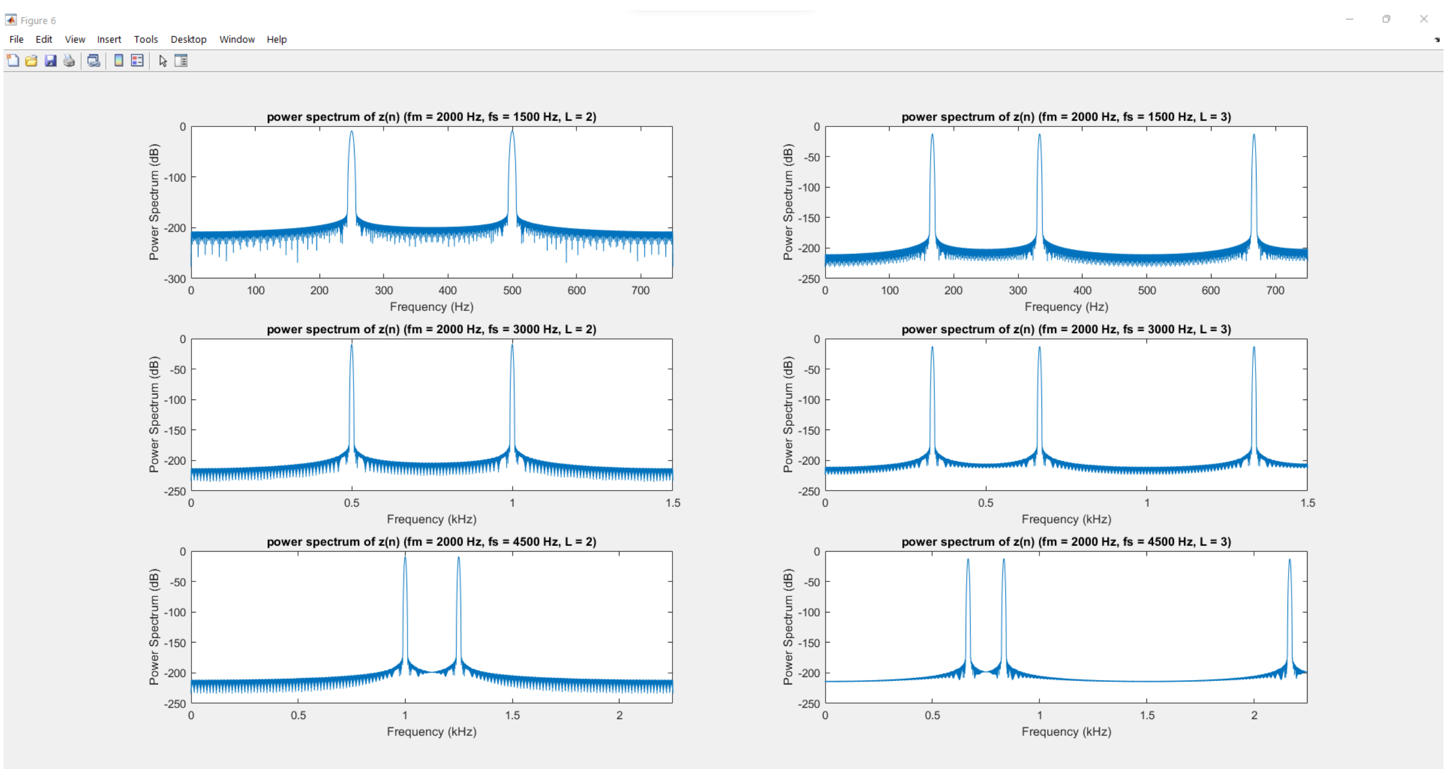


Power Spectrum of $y(n)$ at different sampling frequencies (aliasing can be observed for all sampling frequencies)

Part 3:



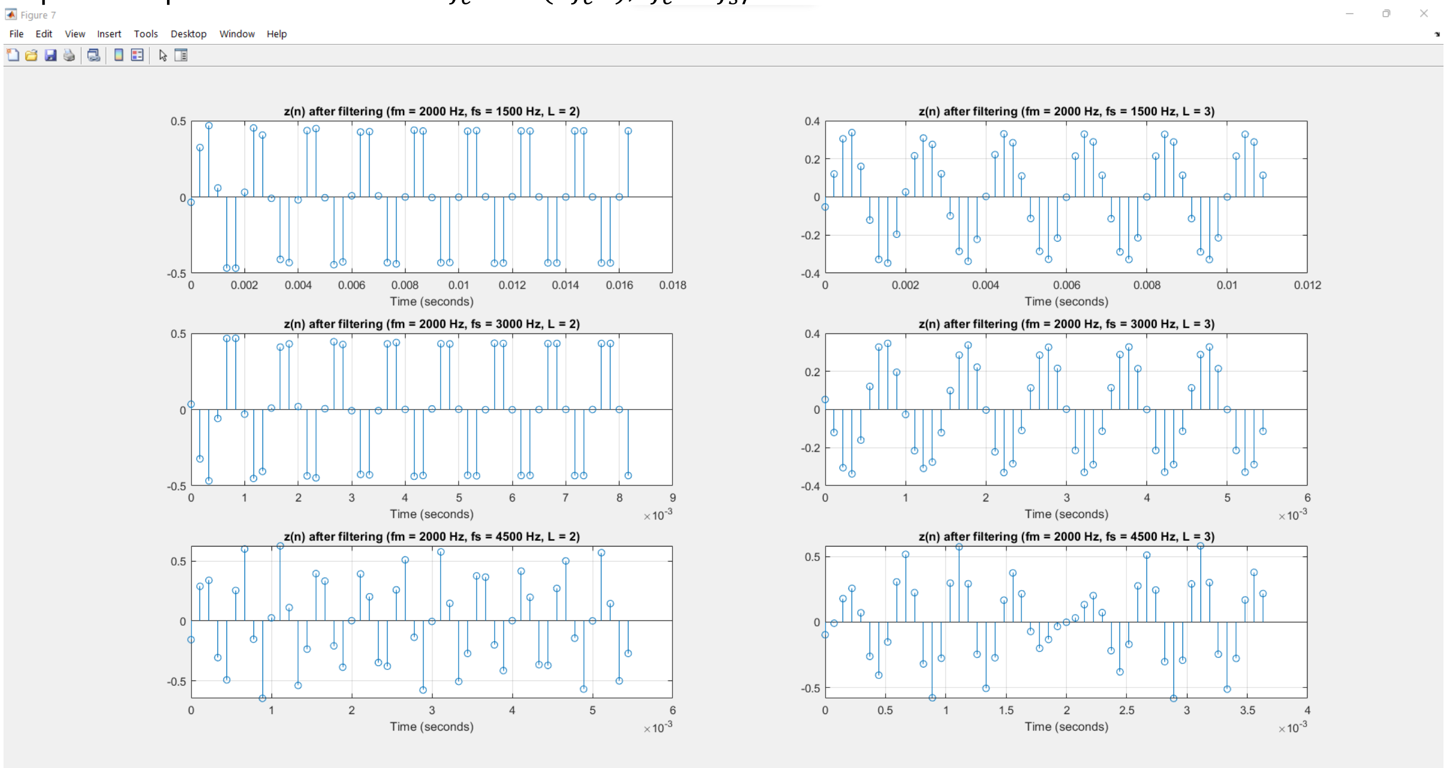
Plot of $z(n)$ at different sampling frequencies



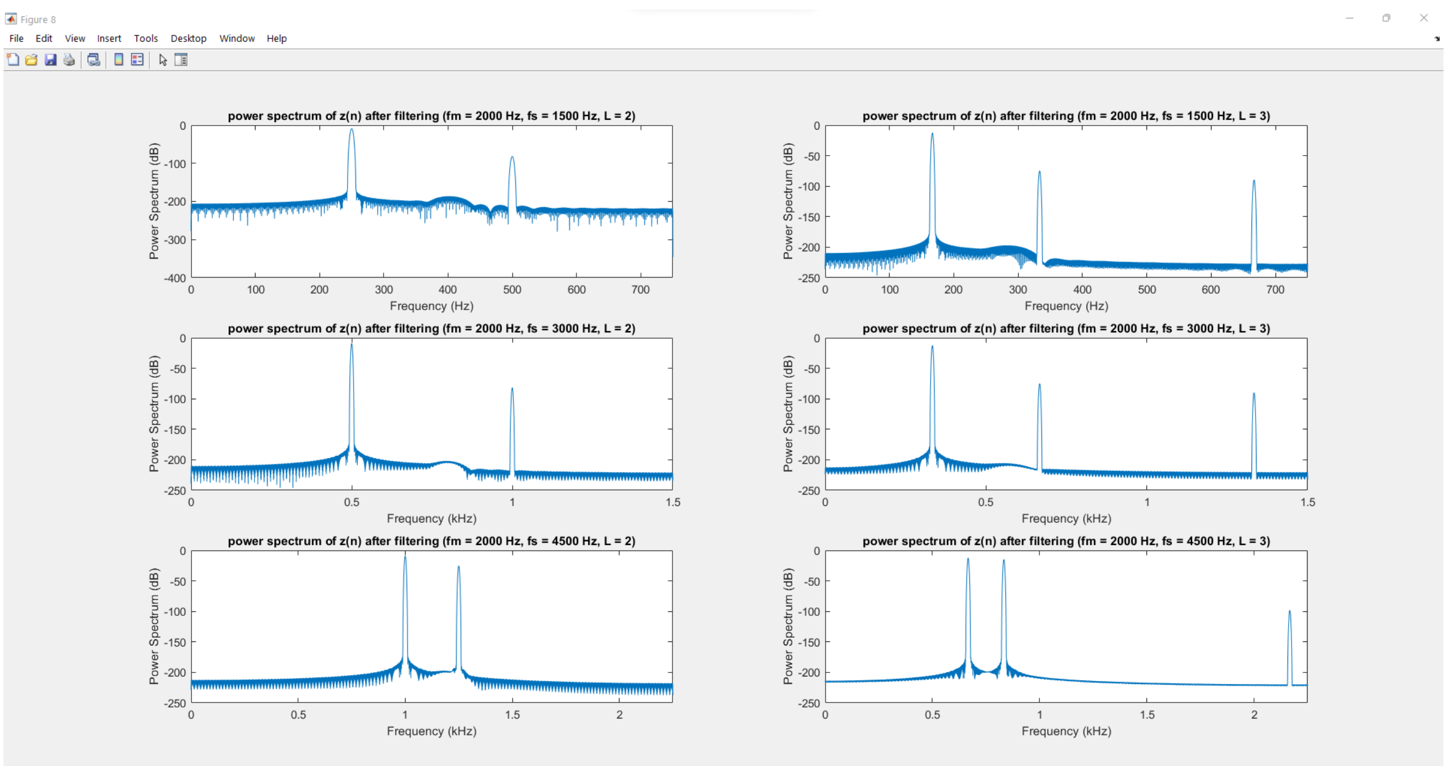
Power Spectrum of $z(n)$ at different sampling frequencies

Part 4:

Impulse response of the filter: $2f_c\, sinc(2f_c n),\ f_c = f_s/2L$



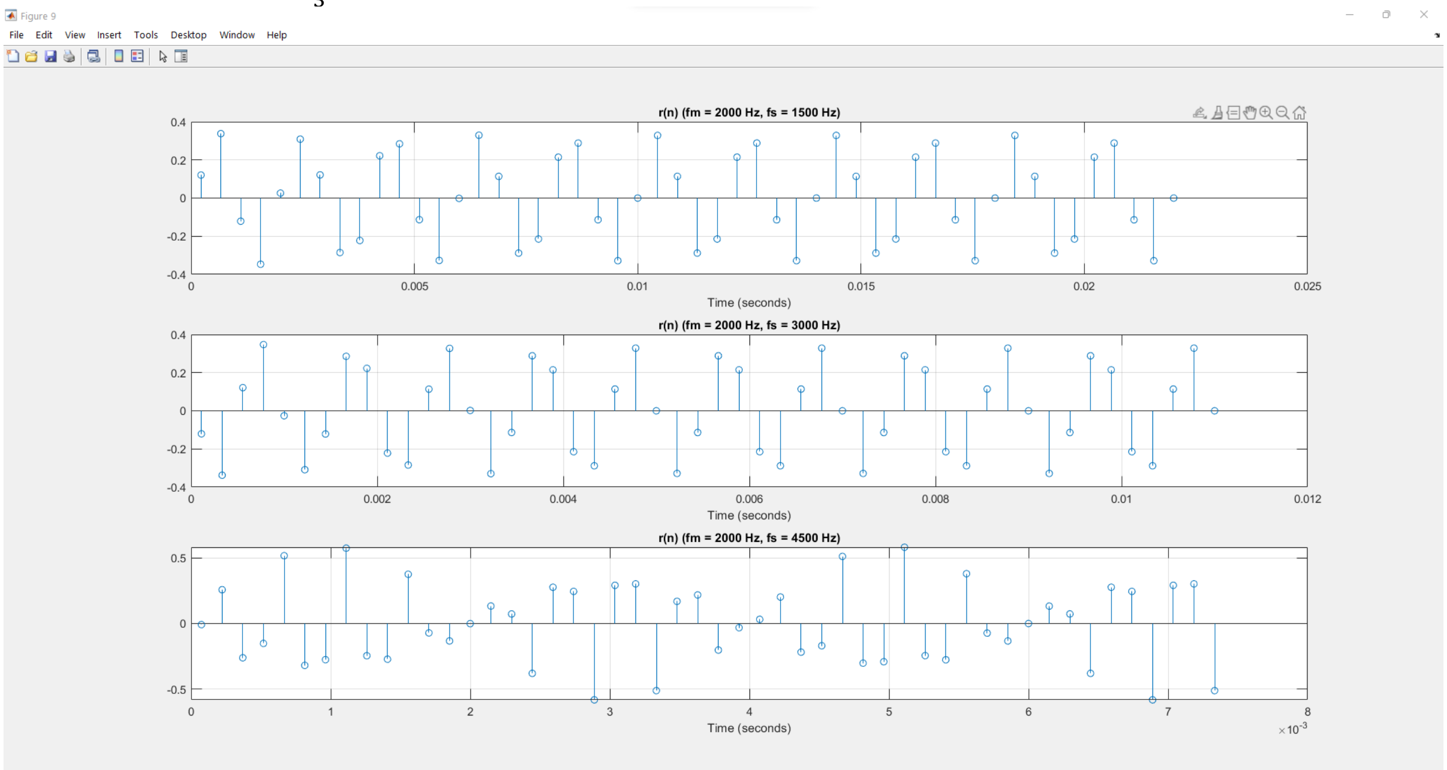
Plot of $z(n)$ after filtering using $sinc$ function



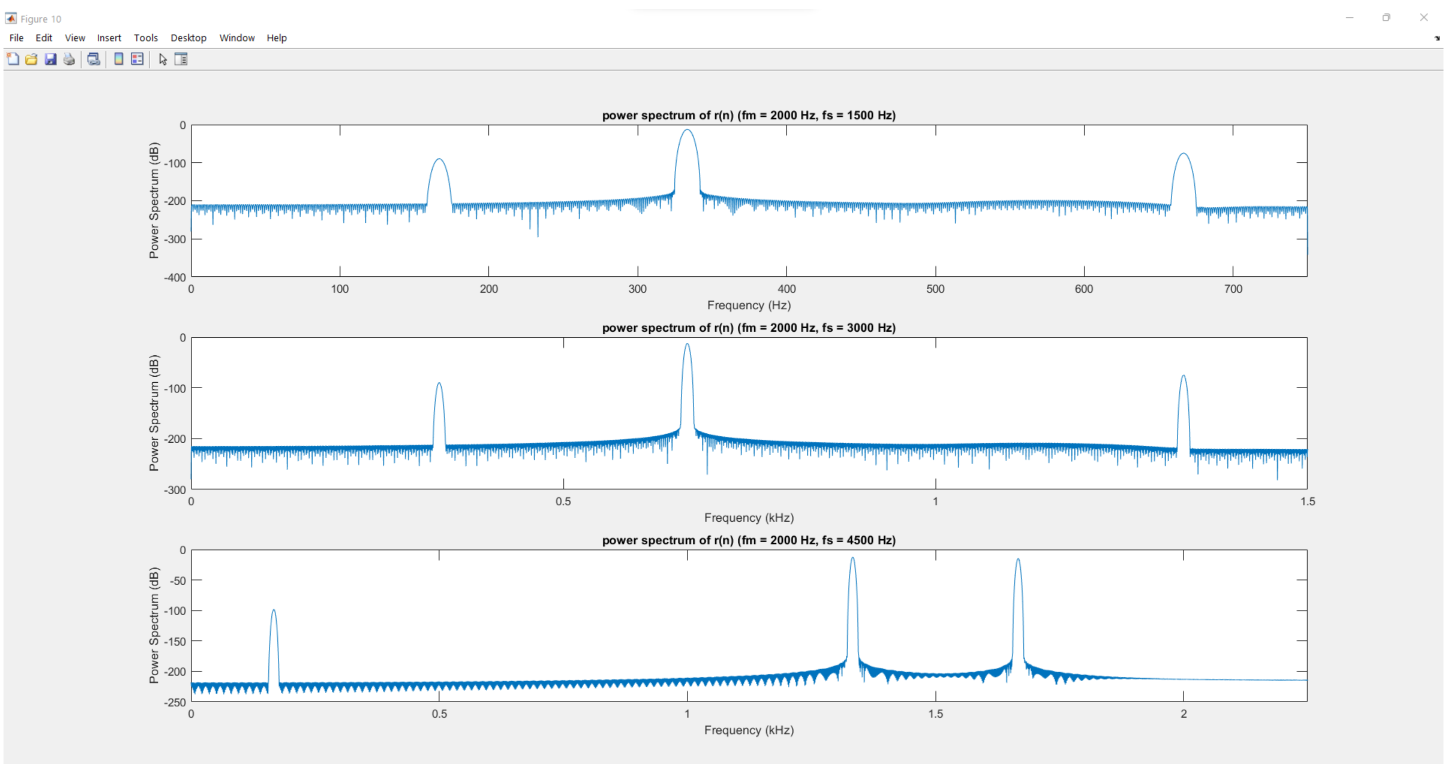
Power Spectrum of $z(n)$ after lowpass filtering

Part 5:

Frequency of $r(n) = \frac{2}{3} f_m$, where f_m, denotes message frequency if there's no aliasing.



Plot of $r(n)$ at different sampling frequencies



Power Spectrum of $r(n)$ at different sampling frequencies (aliasing can be observed for $f_s = 1500\text{ Hz}, 3000\text{ Hz}$)

Conclusion and Observations

1. We can observe the sound and power spectrum of each of the sequences. The pitch of the sound depends on the location of the peak in the power spectrum of the signal. In general, the higher the frequency at which peak is located, higher will be the pitch.

2. Aliasing is observed in case of $x(n)$ for $f_s = 1500\,\mathrm{Hz}, 3000\,\mathrm{Hz}$. In downsampling, aliasing is observed for all sampling frequencies. Finally, in case of $r(n)$, aliasing is observed for $f_s = 1500\,\mathrm{Hz}, 3000\,\mathrm{Hz}$.

3. In the case of downsampling, only one peak is observed in the power spectrum. However, in case of upsampling, the number of peaks is equal to L.

4. To obtain true interpolated signals from upsampled signals, the upsampled signals must be passed through a lowpass filter. We can implement these filters in time-domain using $sinc$ function.

5. After filtering the upsampled signals, we can observe that the magnitude of additional peaks in the power spectrum is greatly reduced.
