EESC6360

Project 1 – Sampling and Sunspots

Josiah W. Smith

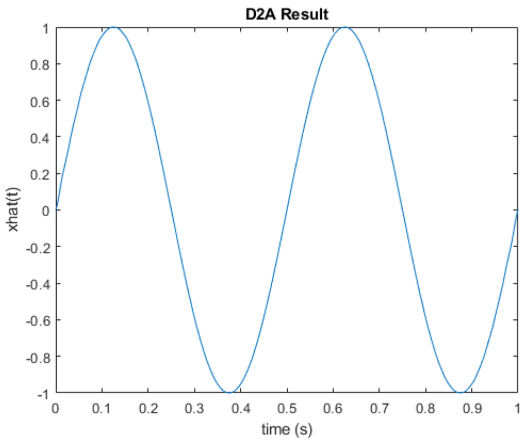
March 7th, 2019

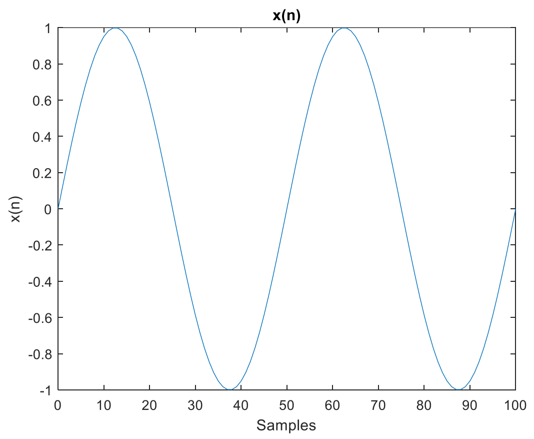
Question 1:

Part A:

```
N = 100;
n = 0:N;
T = 1/100;
f = 2;

x = sin(2*pi*f*n*T);

[xhat,t] = PROJECT1(x,N,T);

plot(t,xhat)
xlabel('time (s)')
ylabel('xhat(t)')
title('D2A Result')
```

```
function [xhat,t] = PROJECT1(x,N,T)
t = 0:T/20:N*T;
n = 0:length(x)-1;
xhat = zeros(1,length(t));
for i = 1:length(t)
   xhat(i) = sum(x.*sinc((t(i)-n*T)/T));
end
end
```

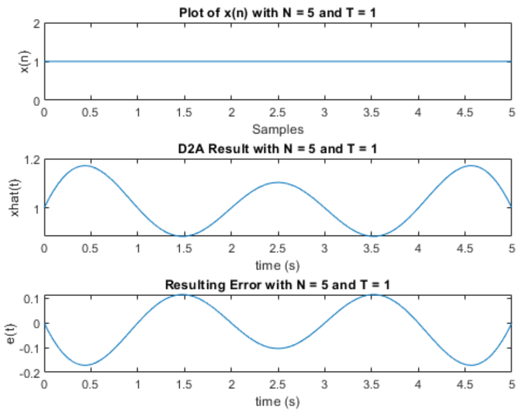


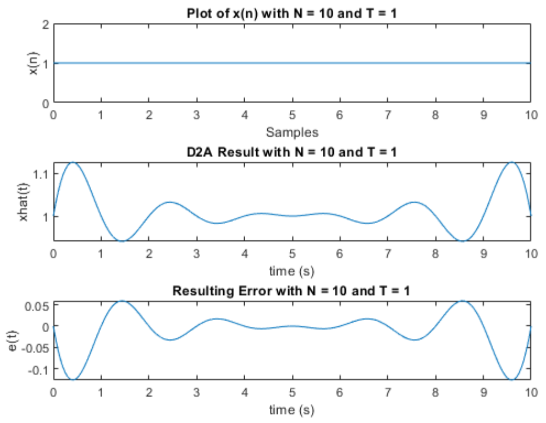


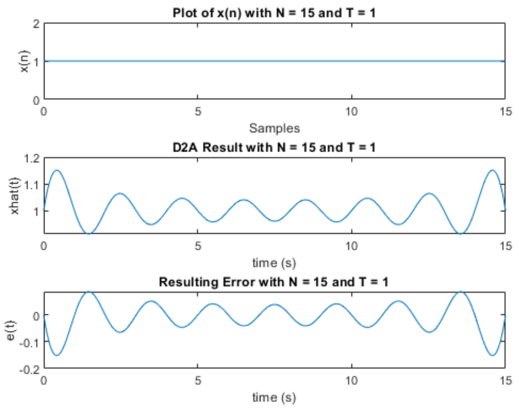
As we can see from D2A result, we see a smooth, accurately reconstructed xhat(t) from the input, discrete signal, x(t)

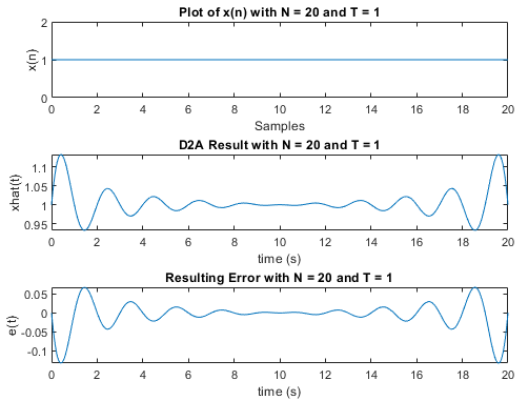
Part B:

```
T = 1;
for N = 5:5:20
    x = ones(1,N+1);
    [xhat,t] = PROJECT1(x,N,T);
    xa = ones(1,length(t));
    e = xa - xhat;
    figure
    subplot(3,1,1);
    plot(0:length(x)-1,x)
    xlabel('Samples')
    ylabel('x(n)')
    title(sprintf('Plot of x(n) with N = %d and T = 1',N))

    subplot(3,1,2);
    plot(t,xhat)
    xlabel('time (s)')
    ylabel('xhat(t)')
    title(sprintf('D2A Result with N = %d and T = 1',N))

    subplot(3,1,3);
    plot(t,e)
    xlabel('time (s)')
    ylabel('e(t)')
    title(sprintf('Resulting Error with N = %d and T = 1',N))
end
```





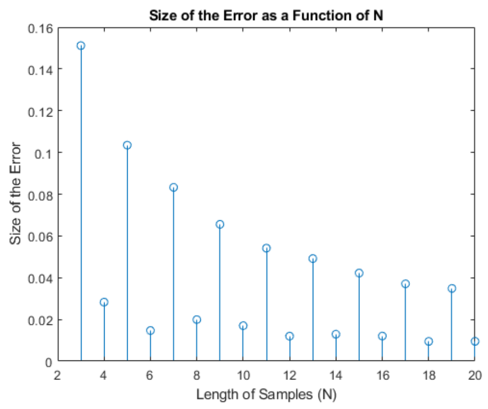




As we can see from the plots above, the reconstructed xhat(t) signal is most smooth near the middle of the time region. At the edges of the segment, we see more error for the reconstructed signals, as we should expect. In theory, if we had an infinite window from which to collect samples, we would see an exact reconstruction of the x(t) signal because the “edges” of the frame would be at positive and negative infinity. The error, as a function of time, appears to be largest at the beginning and end of the time window we are considering. Near the middle, we see a lower error as there are more samples contributing amplitude via the sinc interpolation to more accurately reconstruct the original signal.

Part C:

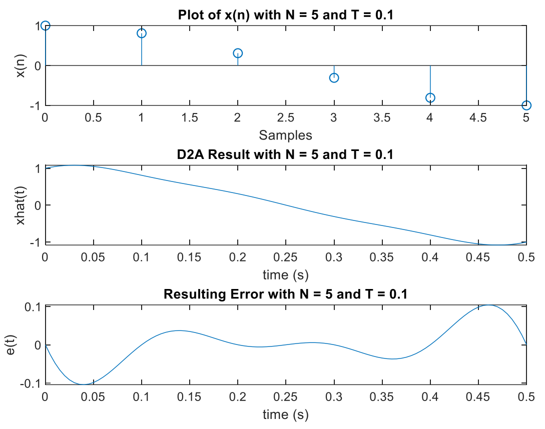
```
T = 1;
e_max = zeros(1,18);
for N = 3:20
    x = ones(1,N+1);
    [xhat,t] = PROJECT1(x,N,T);
    xa = ones(1,length(t));
    e = xa - xhat;
    e_max(N-2) = max(abs(e(round(1/3*length(xa)):round(2/3*length(xa)))));
end
stem(3:length(e_max)+2,e_max)
xlabel('Length of Samples (N)');
ylabel('Size of the Error');
title('Size of the Error as a Function of N');
```

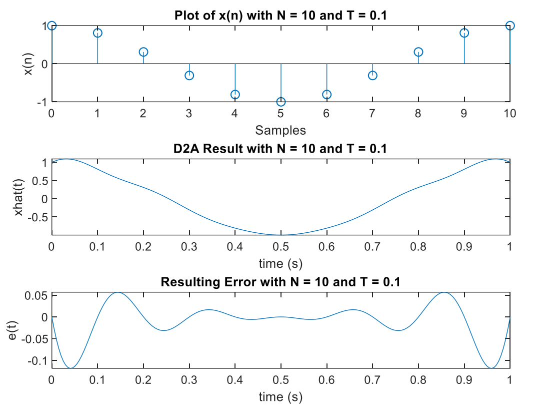


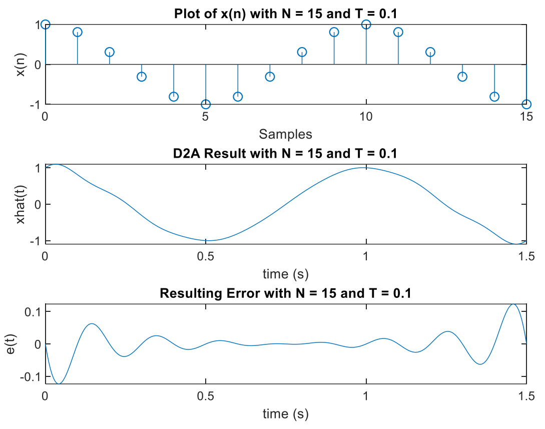
Above is a plot of the size of the error, as defined in the manual, against the size of the frame (N). We notice that the error seems to decrease as N increases. There also appears to be a phenomenon occurring that results in a smaller error for even N versus odd N. However, as a general trend, as we expected, the size of the error decreases as N increases.

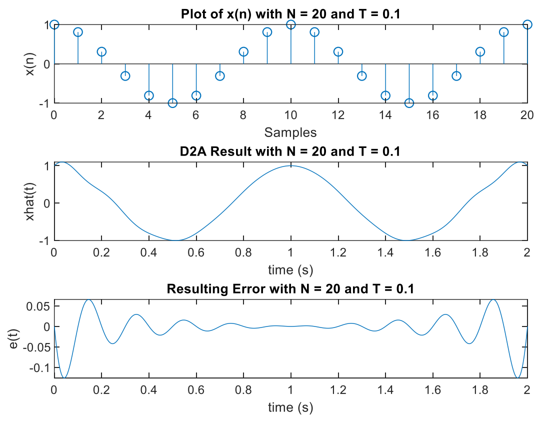
Part D:

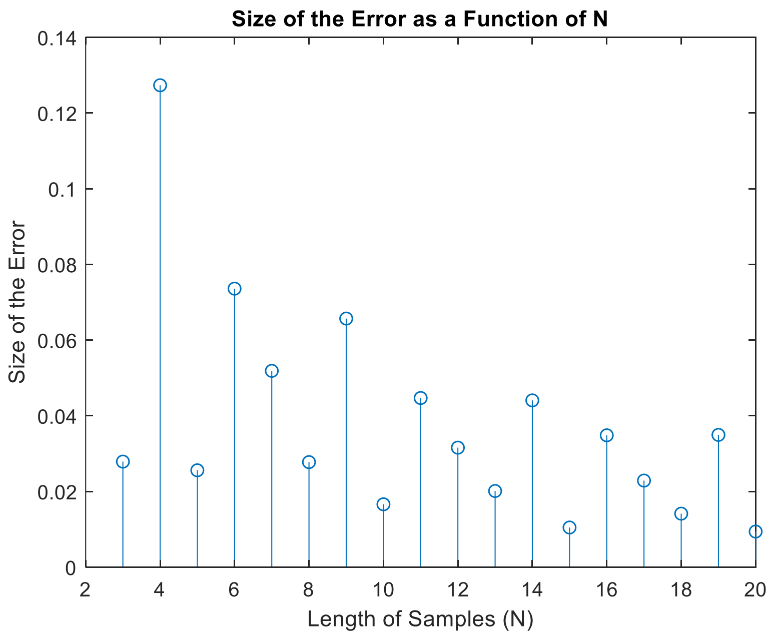
```
for N = 5:5:20
    F = 1;
    T = 0.1;
    n = 0:N;
    Theta = 0;
    x = cos(2*pi*F*T*n + Theta);
    [xhat,t] = PROJECT1(x,N,T);
    xa = cos(2*pi*F*t + Theta);

    e = xa - xhat;
    figure
    subplot(3,1,1)
    stem(0:length(x)-1,x)
    xlabel('Samples')
    ylabel('x(n)')
    title(sprintf('Plot of x(n) with N = %d and T = 1',N))

    subplot(3,1,2)
    plot(t,xhat)
    xlabel('time (s)')
    ylabel('xhat(t)')
    title(sprintf('D2A Result with N = %d and T = 1',N))

    subplot(3,1,3)
    plot(t,e)
    xlabel('time (s)')
    ylabel('e(t)')
    title(sprintf('Resulting Error with N = %d and T = 1',N))
end
e_max = zeros(1,18);
for N = 3:20
    n = 0:N;
    x = cos(2*pi*F*T*n + Theta);
    [xhat,t] = PROJECT1(x,N,T);
    xa = cos(2*pi*F*t + Theta);
    e = xa - xhat;
    e_max(N-2) = max(abs(e(round(1/3*length(xa)):round(2/3*length(xa)))));
end
figure
stem(3:length(e_max)+2,e_max)
xlabel('Length of Samples (N)');
ylabel('Size of the Error');
title('Size of the Error as a Function of N');
```


Plot of x(n) with N = 5 and T = 0.1
x(n)
Samples
D2A Result with N = 5 and T = 0.1
xhat(t)
time (s)
Resulting Error with N = 5 and T = 0.1
e(t)
time (s)


Plot of x(n) with N = 10 and T = 0.1
x(n)
Samples
D2A Result with N = 10 and T = 0.1
xhat(t)
time (s)
Resulting Error with N = 10 and T = 0.1
e(t)
time (s)


Plot of x(n) with N = 15 and T = 0.1
x(n)
Samples
D2A Result with N = 15 and T = 0.1
xhat(t)
time (s)
Resulting Error with N = 15 and T = 0.1
e(t)
time (s)

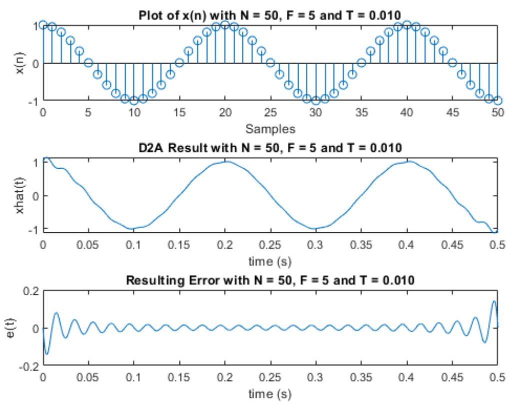

Plot of x(n) with N = 20 and T = 0.1
x(n)
Samples
D2A Result with N = 20 and T = 0.1
xhat(t)
time (s)
Resulting Error with N = 20 and T = 0.1
e(t)
time (s)

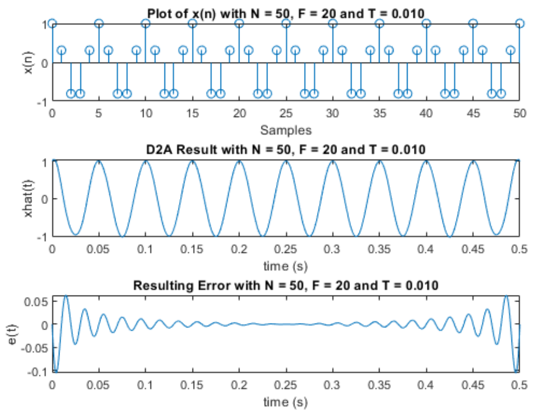

Size of the Error as a Function of N
Size of the Error
Length of Samples (N)

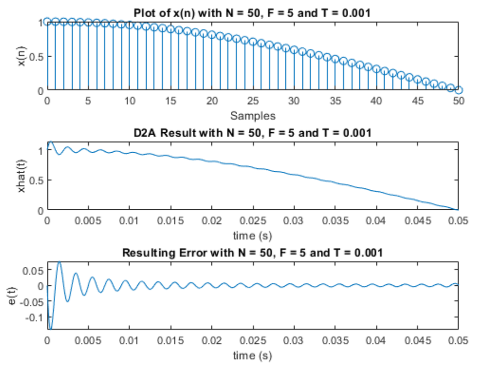
Above are the plots of the discrete signal x(n) compare with the reconstructed xhat(t) and the resulting error for that given frame (N). As expected, we notice a very similar trend to when we used a constant x(t). Again, the error in the middle of the time region taken into consideration is the lowest, with larger errors at the beginning and the end. Additionally, as we increase N, we see a similar result. The even N have a lower size of the error than the odd N; and, the size of the error decreases with respect to N increasing. Compared with part C, we see very similar results with the error in the same range for N between 3 and 20; also, the trend of the size of the error is decreasing as N increases.

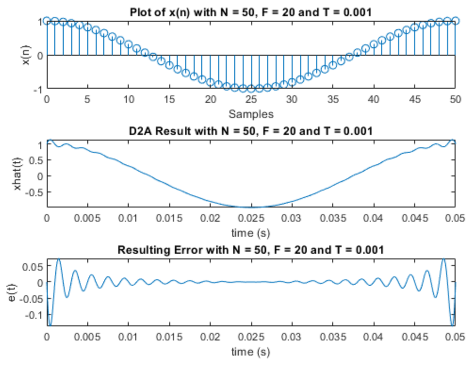
Part E:

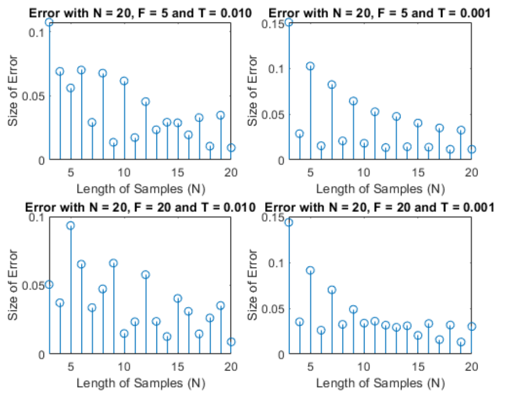
```
for T = [0.01 0.001]
    for F = [5 20]
        N = 50;
        n = 0:50;
        Theta = 0;
        x = cos(2*pi*F*T*n + Theta);
        [xhat,t] = PROJECT1(x,N,T);
        xa = cos(2*pi*F*t + Theta);

        e = xa - xhat;
        figure
        subplot(3,1,1)
        stem(0:length(x)-1,x)
        xlabel('Samples')
        ylabel('x(n)')
        title(sprintf('Plot of x(n) with N = %d, F = %d and T = %1.3f',N,F,T))

        subplot(3,1,2)
        plot(t,xhat)
        xlabel('time (s)')
        ylabel('xhat(t)')
        title(sprintf('D2A Result with N = %d, F = %d and T = %1.3f',N,F,T))

        subplot(3,1,3)
        plot(t,e)
        xlabel('time (s)')
        ylabel('e(t)')
        title(sprintf('Resulting Error with N = %d, F = %d and T = %1.3f',N,F,T))
    end
end
T1 = [0.01 0.001]; F1 = [5 20]; e_max = zeros(1,18); figure
for ii = 1:2
    for jj = 1:2
        for N = 3:20
            n = 0:N;
            T = T1(ii);
            F = F1(jj);
            x = cos(2*pi*F*T*n + Theta);
            [xhat,t] = PROJECT1(x,N,T);
            xa = cos(2*pi*F*t + Theta);
            e = xa - xhat;
            e_max(N-2) = max(abs(e(round(1/3*length(xa)):round(2/3*length(xa)))));
        end
        subplot(2,2,ii+2*(jj-1))
        stem(3:length(e_max)+2,e_max)
        xlim([3,20])
        xlabel('Length of Samples (N)');
        ylabel('Size of Error');
        title(sprintf('Error with N = %d, F = %d and T = %1.3f',N,F,T))
    end
end
```







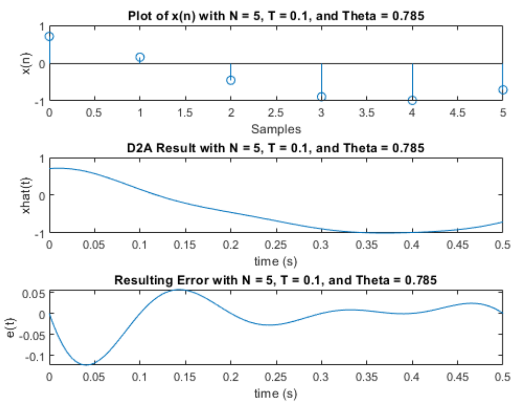


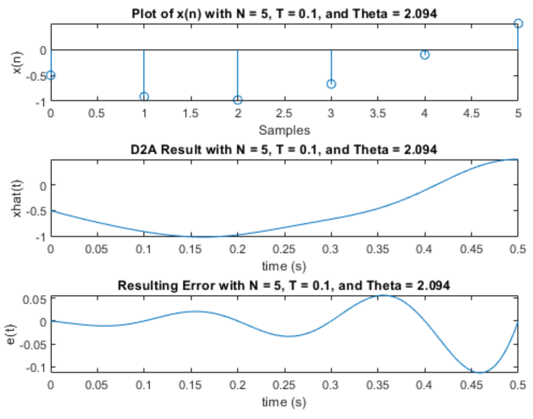


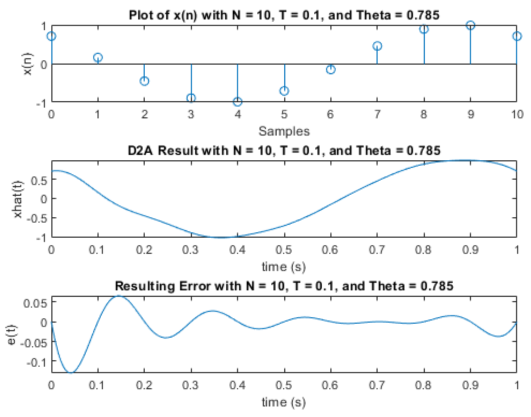
Above are plots of x(n), xhat(t), and the resulting error for multiple values of F and T. Analyzing the last figure above, since we different shapes and amplitudes for the various values of T and F, thus f, we conclude that the size of the error is dependent on the normalized frequency. This is likely because of the dynamic between the size of the frame (N) and the normalized frequency (f). Depending on this relationship, there is a different amount of the wavelength captured by the frame and thus the error will depend on the normalized frequency.

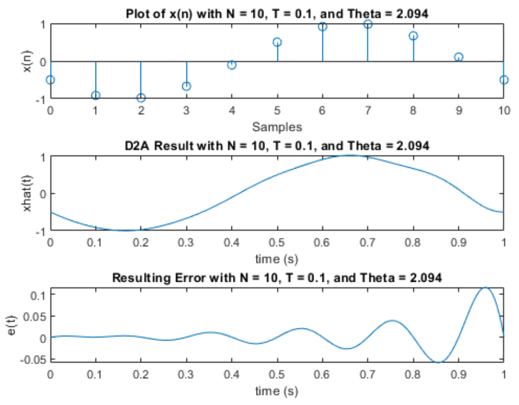
Part F:

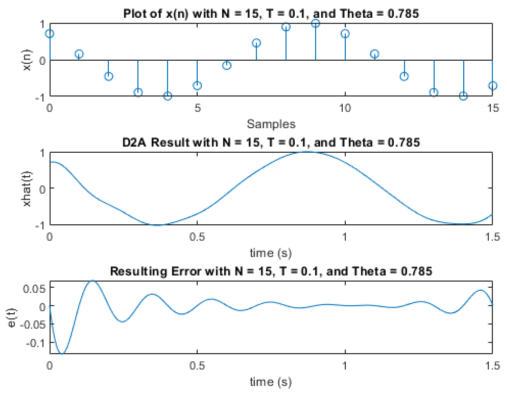
```
for N = 5:5:20
    for Theta = [pi/4 2*pi/3]
        F = 1;
        T = 0.1;
        n = 0:N;
        x = cos(2*pi*F*T*n + Theta);
        [xhat,t] = PROJECT1(x,N,T);
        xa = cos(2*pi*F*t + Theta);

        e = xa - xhat;
        figure
        subplot(3,1,1)
        stem(0:length(x)-1,x)
        xlabel('Samples')
        ylabel('x(n)')
        title(sprintf('Plot of x(n) with N = %d, T = 0.1, and Theta = %1.3f',N,Theta))

        subplot(3,1,2)
        plot(t,xhat)
        xlabel('time (s)')
        ylabel('xhat(t)')
        title(sprintf('D2A Result with N = %d, T = 0.1, and Theta = %1.3f',N,Theta))

        subplot(3,1,3)
        plot(t,e)
        xlabel('time (s)')
        ylabel('e(t)')
        title(sprintf('Resulting Error with N = %d, T = 0.1, and Theta = %1.3f',N,Theta))
    end
end
e_max = zeros(1,18);
for Theta = [pi/4 2*pi/3]
    for N = 3:20
        n = 0:N;
        x = cos(2*pi*F*T*n + Theta);
        [xhat,t] = PROJECT1(x,N,T);
        xa = cos(2*pi*F*t + Theta);
        e = xa - xhat;
        e_max(N-2) = max(abs(e(round(1/3*length(xa)):round(2/3*length(xa)))));
    end
    figure
    stem(3:length(e_max)+2,e_max)
    xlabel('Length of Samples (N)');
    ylabel('Size of the Error');
    title(sprintf(' Error with N = %d, T = 0.1, and Theta = %1.3f',N,Theta))
end
```

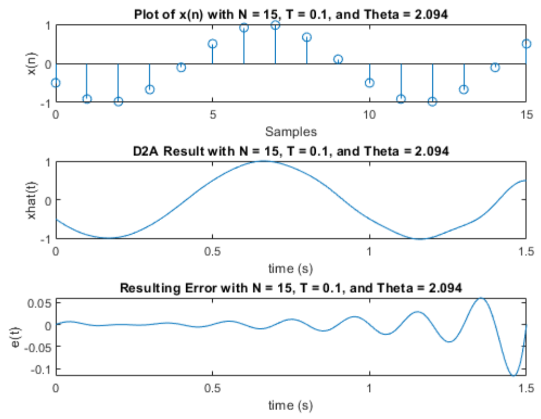


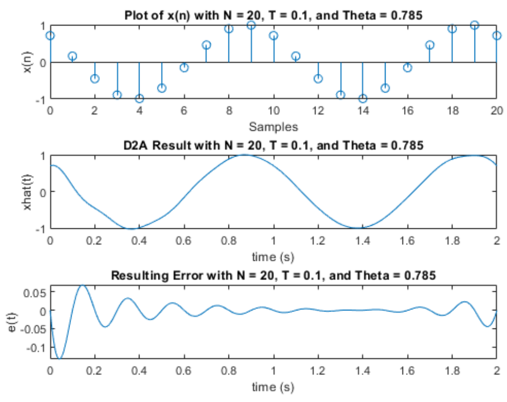


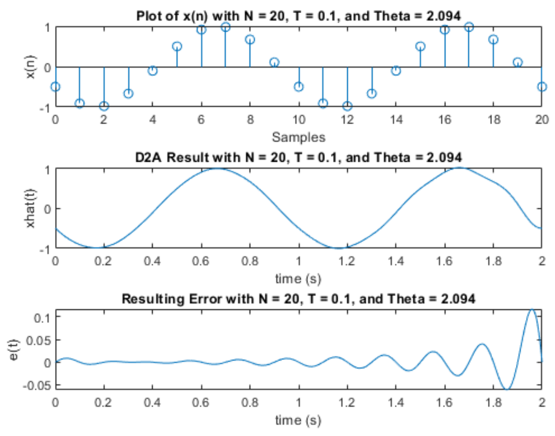


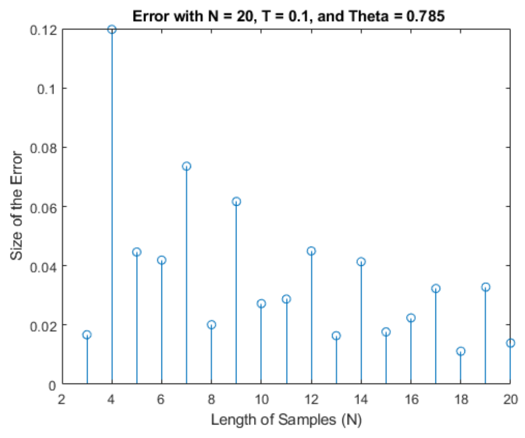


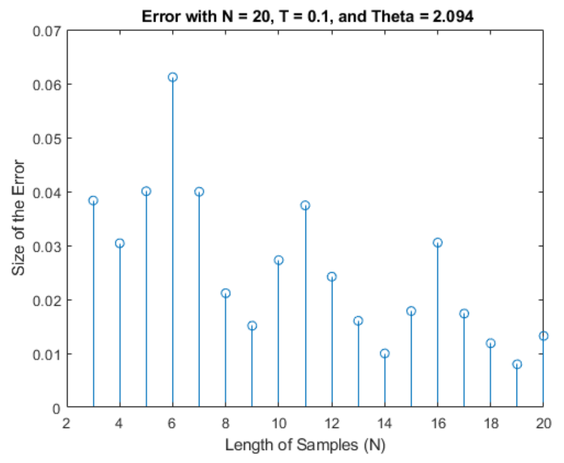












The figures above show x(n), xhat(t), and the resulting error for various values of N and θ, followed by two plots of the size of the error for two different values of θ. Again, we see differences in the size of the error from the last two plots with θ= π/4 and θ= 2π/3. Additionally, we compare this with the size of the error from part E, we see they all have different sizes of error. From this, we can conclude that the size of the error is dependent on the phase (θ). Similar to the normalized frequency, the phase (θ) results in different portions of the sinusoid being captured by the frame and thus different error depending on θ.

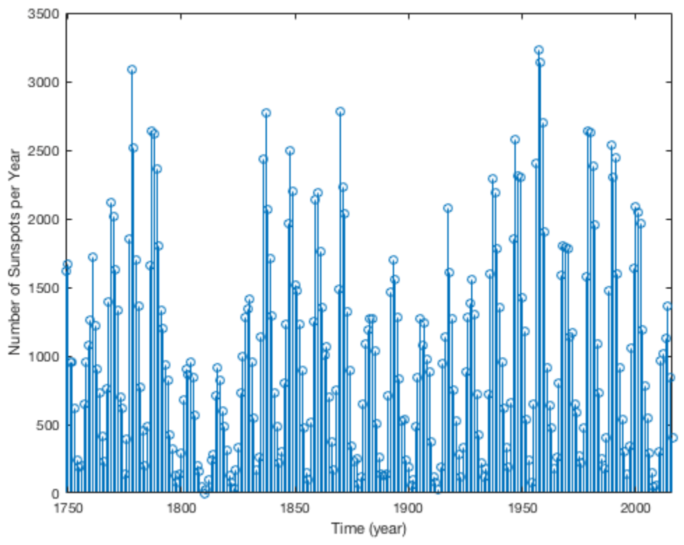
Question 2:

```
        fileID = fopen('SN_m_tot_V2.0.txt','r');
        formatSpec = '%d %d %f %f %f %d';
        sizeA = [6 Inf];
        A = fscanf(fileID,formatSpec,sizeA);
```

Part A:

```
num_ss_month = A(4,:);
year = linspace(1749,2016,268);
num_ss_year = zeros(1,268);
for i = 1:12:3204
    num_ss_year(A(1,i)-1748) = sum(A(4,i:i+11));
```

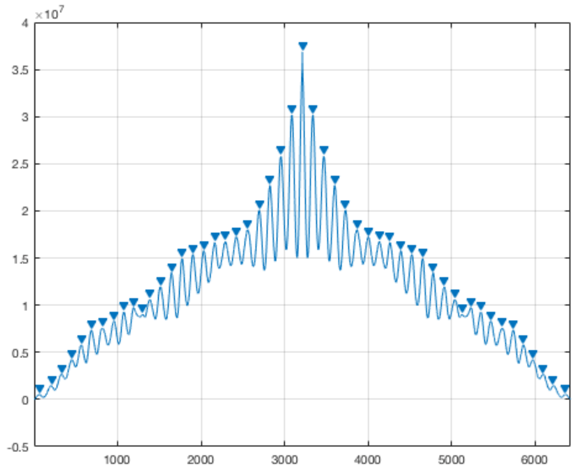
```
end

num_ss_year(A(1,3205)-1748) = sum(A(4,3205:3213));
stem(year,num_ss_year)
xlabel('Time (year)')
ylabel('Number of Sunspots per Year')
xlim([1749,2016])
```



Graph of the number of sunspots per year

Part B:

```
Rnn = xcorr(num_ss_month);
figure
plot(Rnn)
findpeaks(Rnn,'MinPeakDistance',72)
[~,locsn]=findpeaks(Rnn,'MinPeakDistance',72);
Tn=mean(diff(locsn)); % Tn = 125.76 months
```

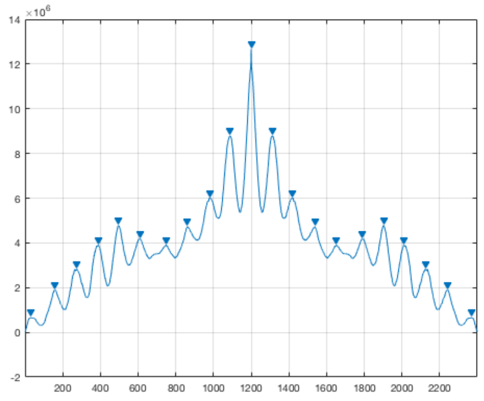


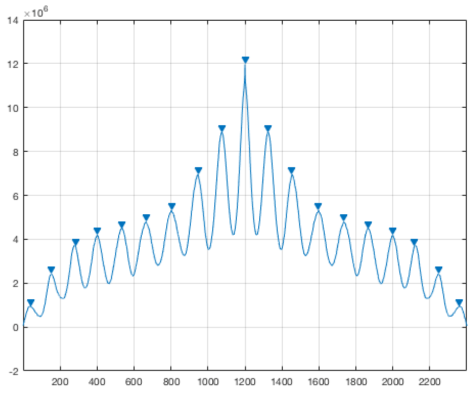
Above is a plot of the autocorrelation of the sunspots with the peaks denoted with triangles. We find the peaks and average the differences between the peaks to find the period of the sunspots in terms of months. We found that the period was approximately 125.76 months or 10.48 years.

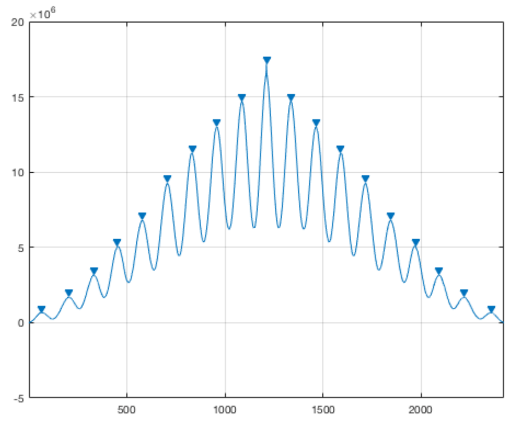
Part C:

% 1749-1848

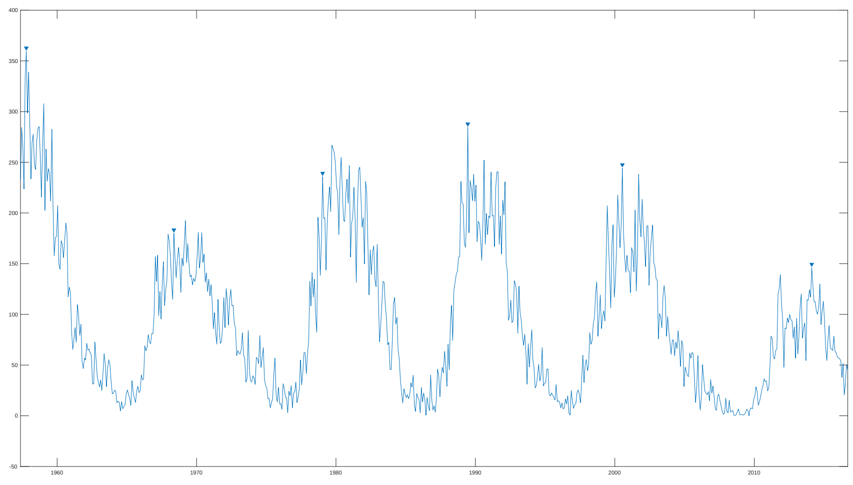
```
Rnn1749_1848 = xcorr(num_ss_month(1:1200));
figure
plot(Rnn1749_1848)
findpeaks(Rnn1749_1848,'MinPeakDistance',72)
[~,locsn]=findpeaks(Rnn1749_1848,'MinPeakDistance',72);
Tn1749_1848=mean(diff(locsn)); % Tn = 116.80 months

% 1849-1948

Rnn1849_1948 = xcorr(num_ss_month(1201:2400));
figure
plot(Rnn1849_1948)
findpeaks(Rnn1849_1948,'MinPeakDistance',72)
[~,locsn]=findpeaks(Rnn1849_1948,'MinPeakDistance',72);
Tn1849_1948=mean(diff(locsn)); % Tn = 128.89 months

% 1915-2015

Rnn1915_2015 = xcorr(num_ss_month(1993:3204));
figure
plot(Rnn1915_2015)
findpeaks(Rnn1915_2015,'MinPeakDistance',72)
[~,locsn]=findpeaks(Rnn1915_2015,'MinPeakDistance',72);
Tn1915_2015=mean(diff(locsn)); % Tn = 127.33 months
```







In the three figures above, we plot the autocorrelation of the sunspots for the three different time intervals. Using the findpeaks() function in MATLAB, we place the triangles as shown at each peak in the autocorrelation. From the average difference among these peaks, we determine the periodicity of the original sunspot signal. We can see that the period between 1749-1848 is about 116.8 months. Between 1849-1948 the average period is 128.89 months. Finally, the period from 1915 and 2015 is 127.3 months. From these results, it appears that the period of the sunspots has increased since 1748 until 2015.



Part D:

As this figure shows, the most recent peak occurred in February 2014. If the average period of sunspot peaks is approximately 128 months (10 years 8 months), the next peak is expected to be around October 2024. Time will tell, put perhaps in the Fall of 2024, the DSP I class will discover that the next peak in sunspots will be in October of that year.