ECEN 220 Lab 4

Niels Clayton
300437590

September 26, 2019

---

1 Computing the DTFT in MATLAB

Below is the DTFT function written to compute figures 1 and 2:

```
function [X] = dtft(x, n, omega)
    V = exp(-1j*omega*n');
    X = V*x;
end
```

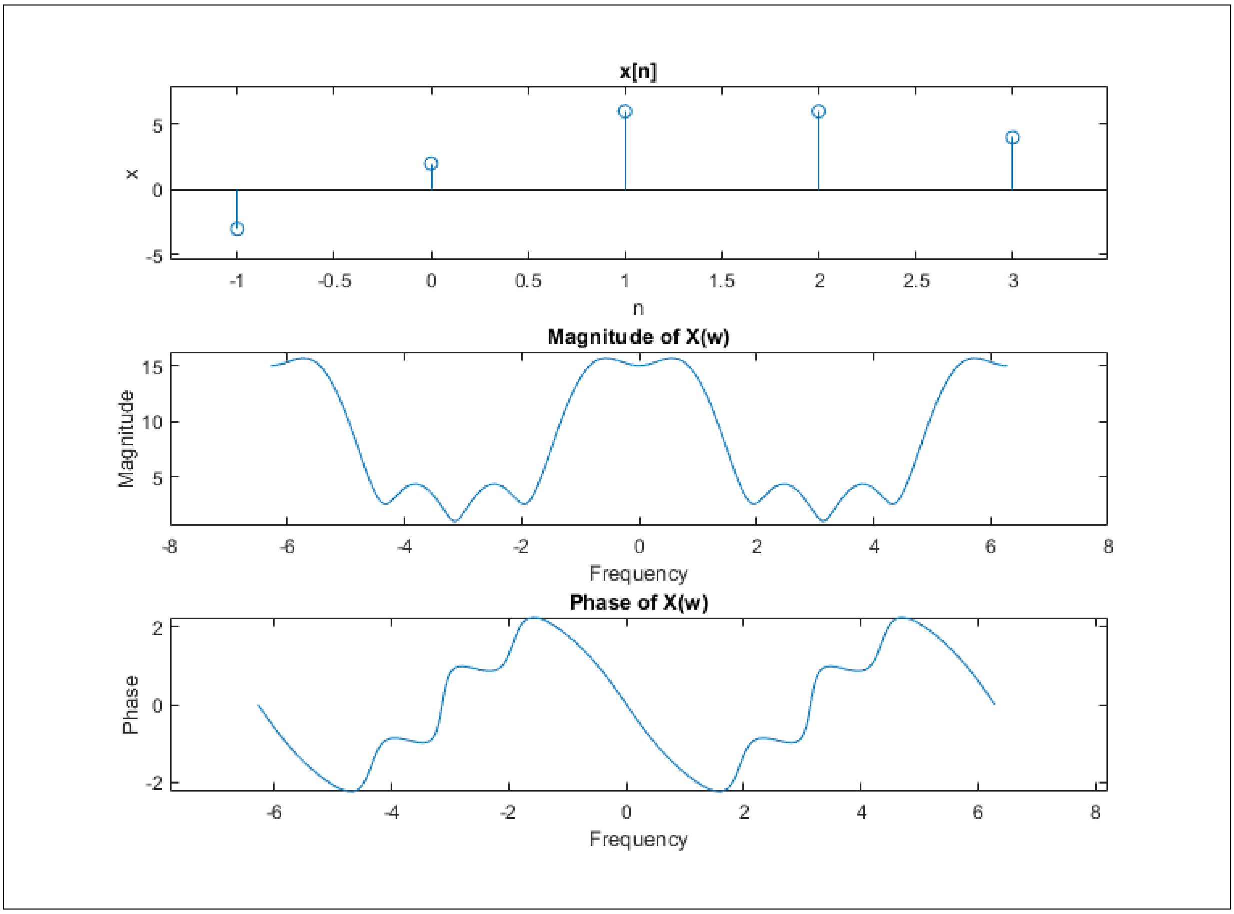

Figure 1: DTFT of signal $x[n]$

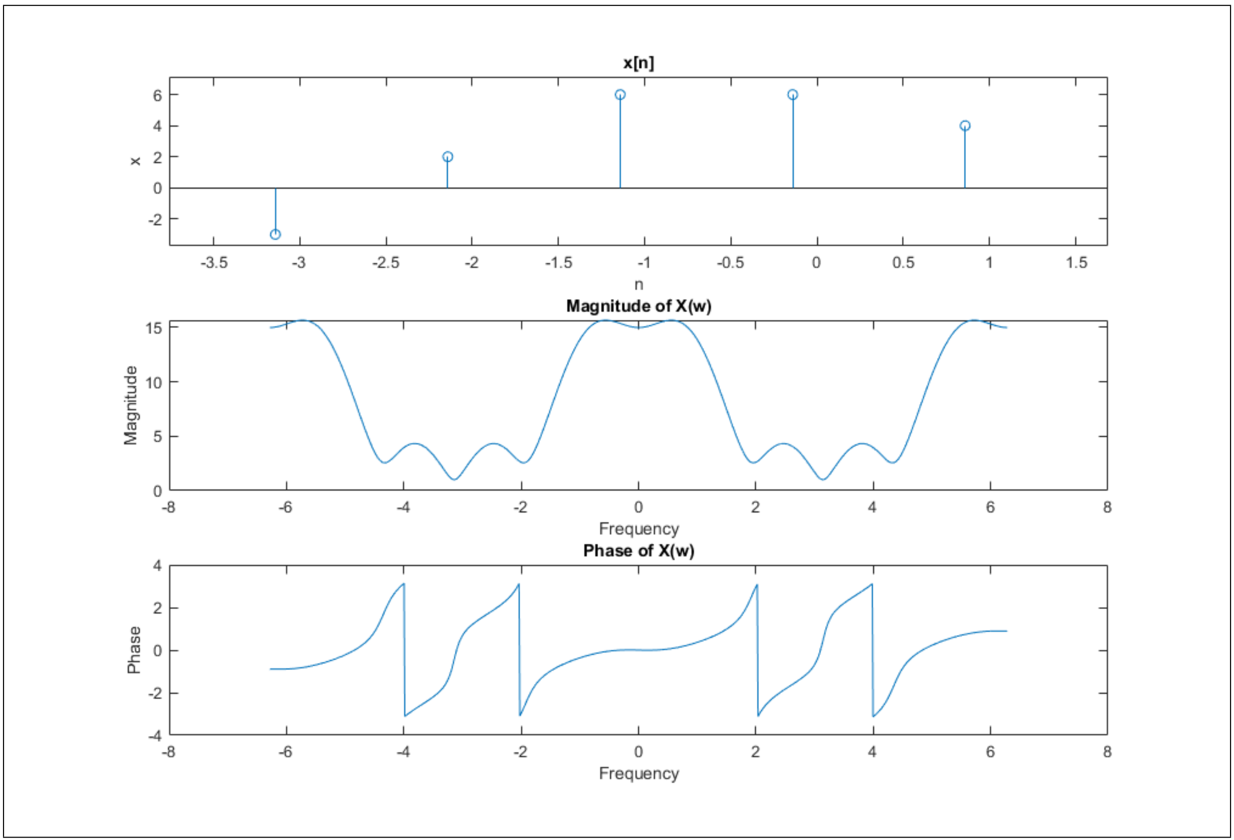

Figure 2: DTFT of signal $x[n+\pi]$

By changing the position of $n = 0$ we are shifting in the time domain, which will lead to a change in the phase in the frequency domain. This can be seen in the differences between figured 1 which is the DTFT of $x[n]$ and figure 2 which is the DTFT of $x[n+\pi]$.

2 Inverse DTFT in MATLAB

Below is the inverse DTFT function written to compute figures 1 and 2:

```
function [x] = invdtft(X, n, omega)
    V = exp(-1j*omega*n');
    x = V\X;
end
```

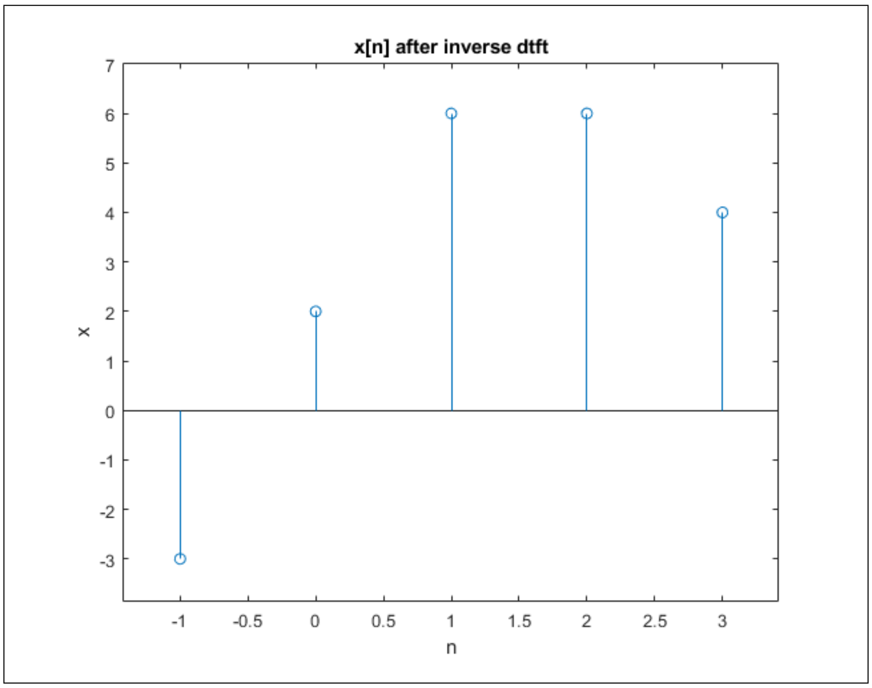

Figure 3: Inverse DTFT of $X(e^{j\omega})$

When taking the inverse discrete time Fourier transform of $X(e^{j\omega})$, it can be seen in figure 3 that the output $x[n]$ is exactly the same as the original signal. Within MATLAB however, every point of $x[n]$ is now considered to be a complex number with an imaginary component of $0i$. This is due to the nature of MATLAB functions, where if there are computations with complex numbers then the output will always be complex, even it it is solely real.

3 Filters

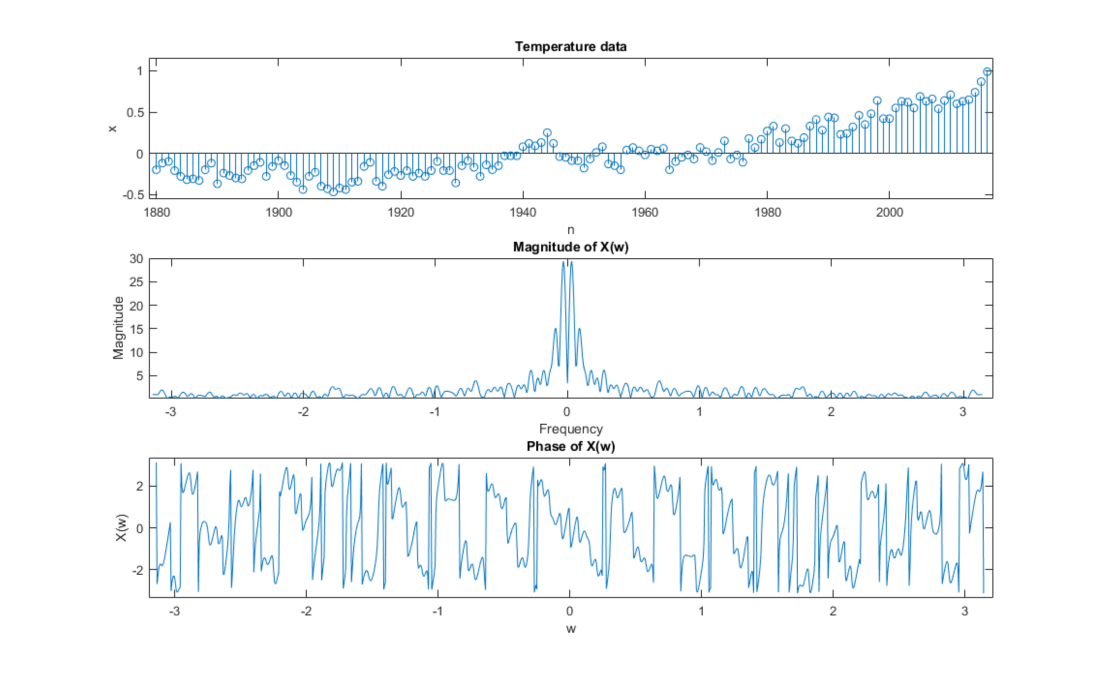

Figure 4: DTFT of temperature data

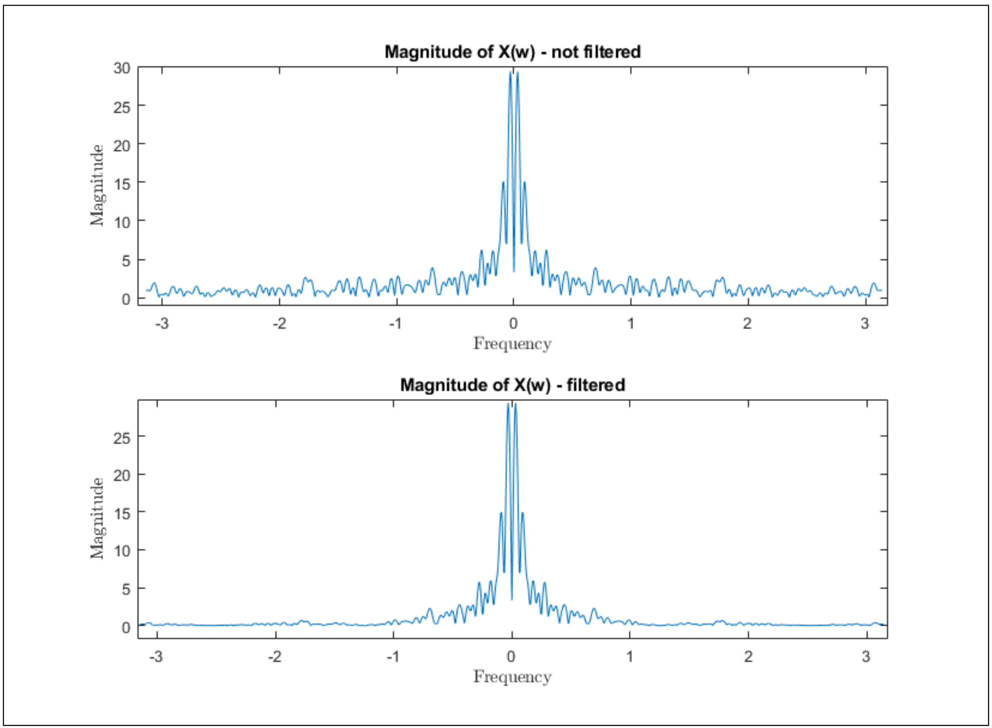

Figure 5: Comparison of un-filtered and filtered data

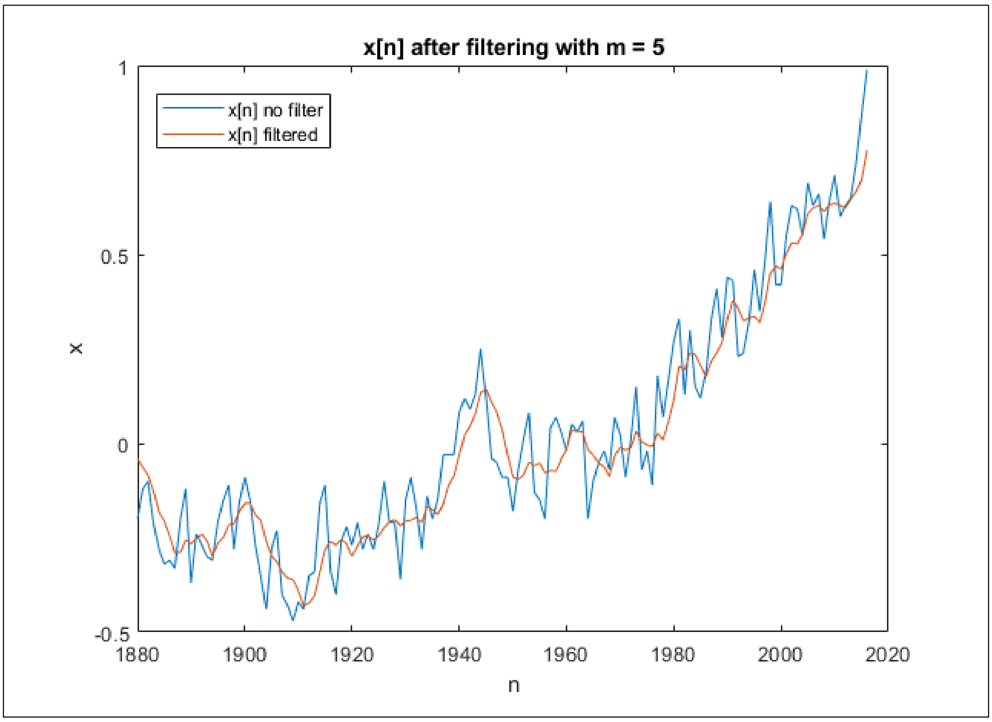

Figure 6: DTFT of temperature data

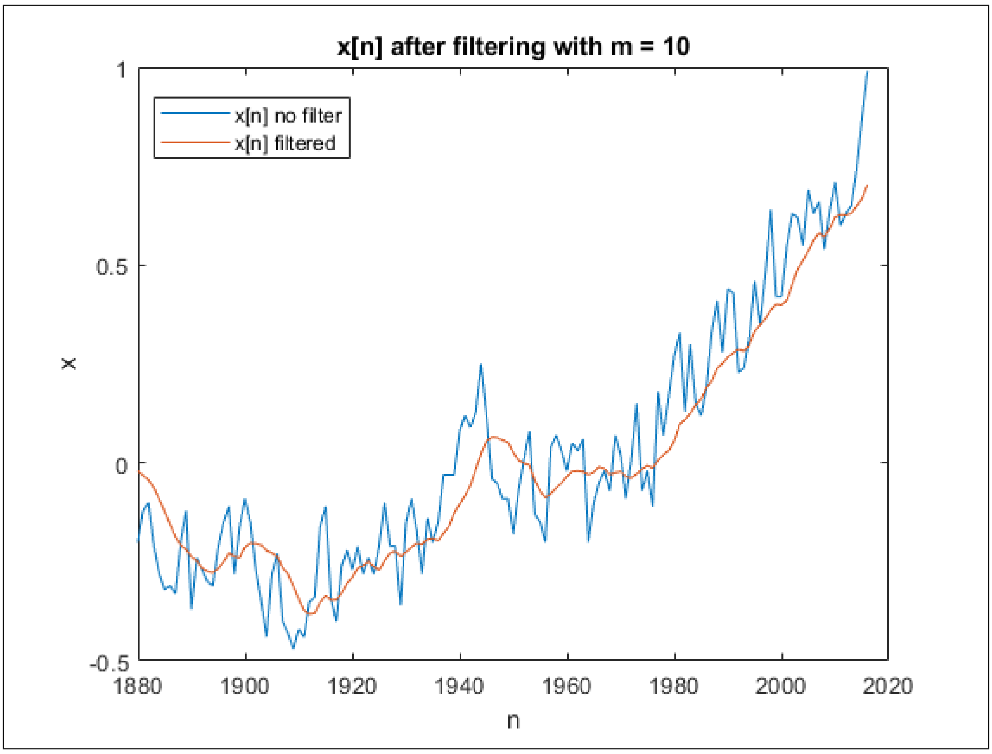

Figure 7: Comparison of un-filtered and filtered data

It can be seen that as the value of m increases, the amount of averaging done to the function increases as well.

4 Extra - High-pass & Low-pass filters

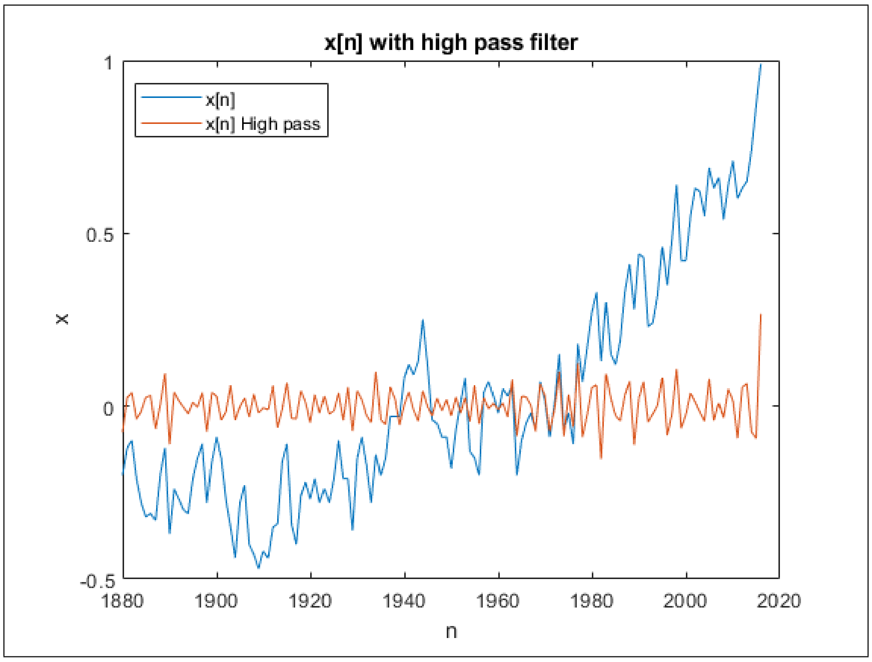

Figure 8: DTFT of temperature data

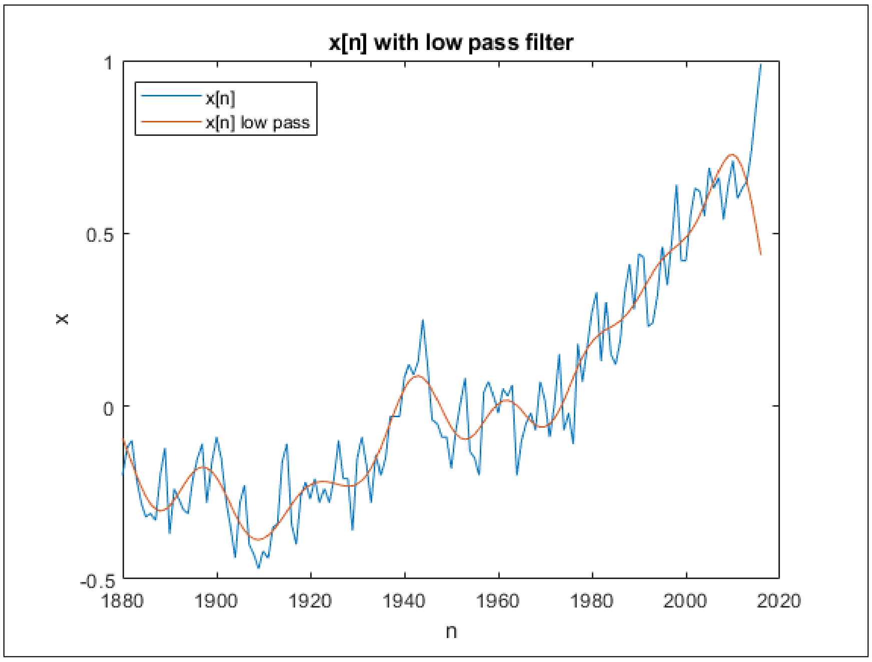

Figure 9: Comparison of un-filtered and filtered data

When passing $X(e^{j\omega})$ through a high pass filter, it can be seen in figure 8 that only the high frequency oscillations are left over, and the low frequency oscillations are removed. this means that the low frequency oscillations are responsible for the overall upwards trend in the data. When passing $X(e^{j\omega})$ through a low pass filter, this upwards trend in temperature can more distinctly be recognised.

Matlab Code

```matlab
%% 1: dtft
clc, clear

M = 1000;
k = (0 : M)';
omega_0 = -2*pi;
omega_M = 2*pi;
omega = omega_0 + (omega_M-omega_0)*(k/M);

x = [-3,2,6,6,4]';
n = (0:length(x)-1)'-pi;
X = dtft( x, n, omega);

subplot(3, 1, 1);
stem(n, x);
title('x[n]')
xlabel('n')
ylabel('x')

subplot(3,1,2);
plot(omega,abs(X));
title("Magnitude of X(w)")
xlabel('Frequency')
ylabel('Magnitude')

subplot(3,1,3);
plot(omega,angle(X));
title("Phase of X(w)")
xlabel('Frequency')
ylabel('Phase')

%% 2: Inverse dtft
clc

x1 = invdtft(X, n, omega)
stem(n, x1);
title('x[n] after inverse dtft')
xlabel('n')
ylabel('x')


%% 3a: dtft of Temperature data
clc, clear

data = importdata('Temperature.txt');
n = data(:, 1);
x = data(:, 2);

omega_0 = -pi;
omega_M = pi;
M = 1000;
k = (0 : M)';
```

```
omega = omega_0 + (omega_M-omega_0)*(k/M);

X = dtft( x, n, omega);

subplot(3, 1, 1);
stem(n, x);
title("Temperature data")
xlabel('n')
ylabel('x')

subplot(3,1,2);
plot(omega,abs(X));
title("Magnitude of X(w)")
xlabel('Frequency')
ylabel('Magnitude')

subplot(3,1,3);
plot(omega,angle(X));
title("Phase of X(w)")
xlabel('w')
ylabel('X(w)')



%% 3: moving average filter

m = 10;
Hav = ((1/m).*exp(-1j.*omega.*((m-1)./2)).*((sin(omega.*m./2))./(sin(
    omega./2))));
Hav((M/2) +1) = 1;

Y = Hav.*X;
y = invdtft(Y, n, omega);

figure(1)
plot(n,x, n,y)
title("x[n] after filtering with m = 10")
xlabel('n')
ylabel('x')
legend('x[n] no filter', 'x[n] filtered');

figure(2)

subplot(2,1,1);
plot(omega,abs(X));
title("Magnitude of X(w) - not filtered")
xlabel('Frequency','interpreter','latex')
ylabel('Magnitude','interpreter','latex')

subplot(2,1,2);
plot(omega,abs(Y));
title("Magnitude of X(w) - filtered")
xlabel('Frequency','interpreter','latex')
ylabel('Magnitude','interpreter','latex')
```

```
%% extra
clc

wc = pi/2;
high_filter = (abs(omega) > wc);

Y = high_filter.*X;
y = invdtft(Y, n, omega);

figure(1)
plot(n,x, n,y)
title("x[n] with high pass filter")
xlabel('n')
ylabel('x')
legend('x[n]', 'x[n] high pass');

wc = pi/8;
low_filter = (abs(omega) < wc);
Y = low_filter.*X;
y = invdtft(Y, n, omega);

figure(2)
plot(n,x, n,y)
title("x[n] with low pass filter")
xlabel('n')
ylabel('x')
legend('x[n]', 'x[n] low pass');
```