

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:

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

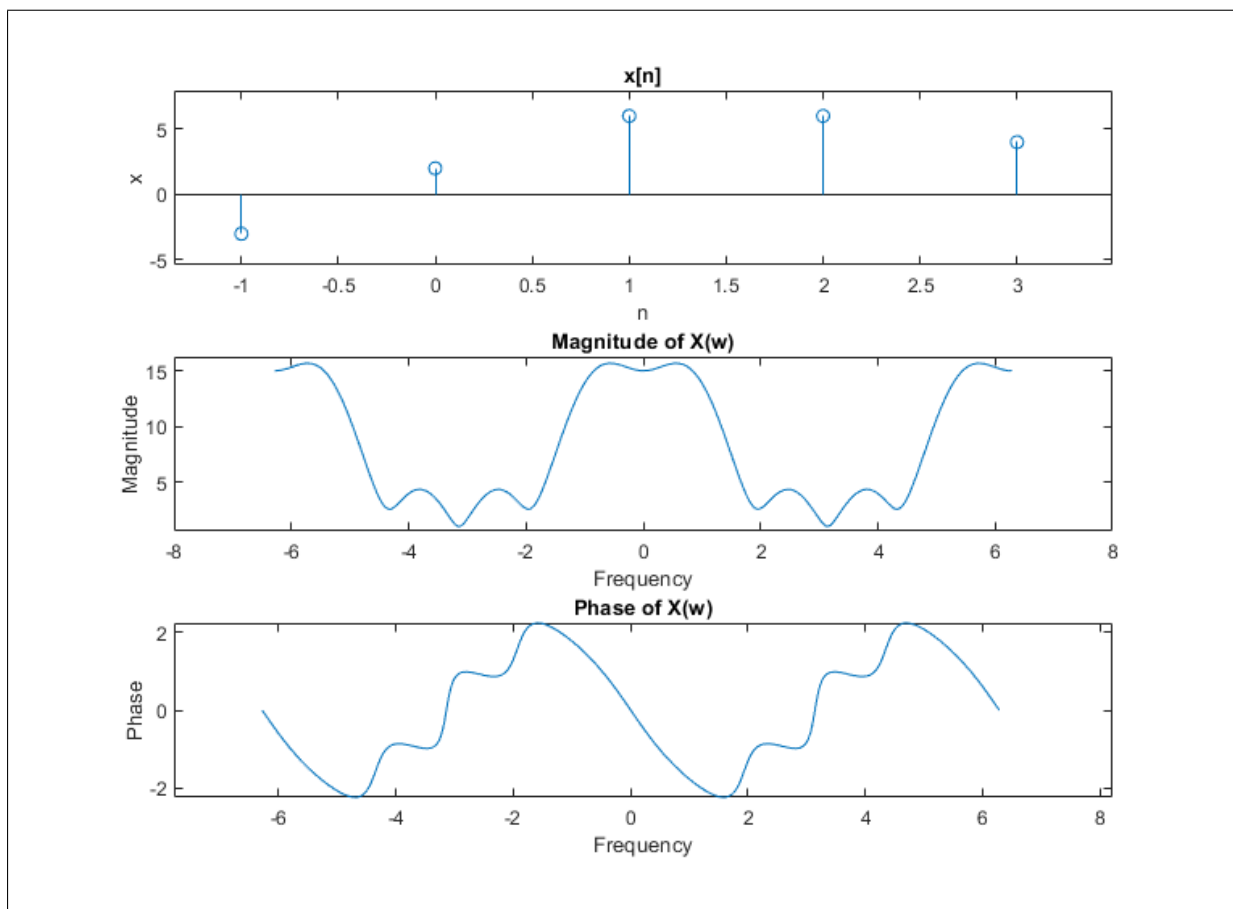


Figure 1: DTFT of signal $x[n]$

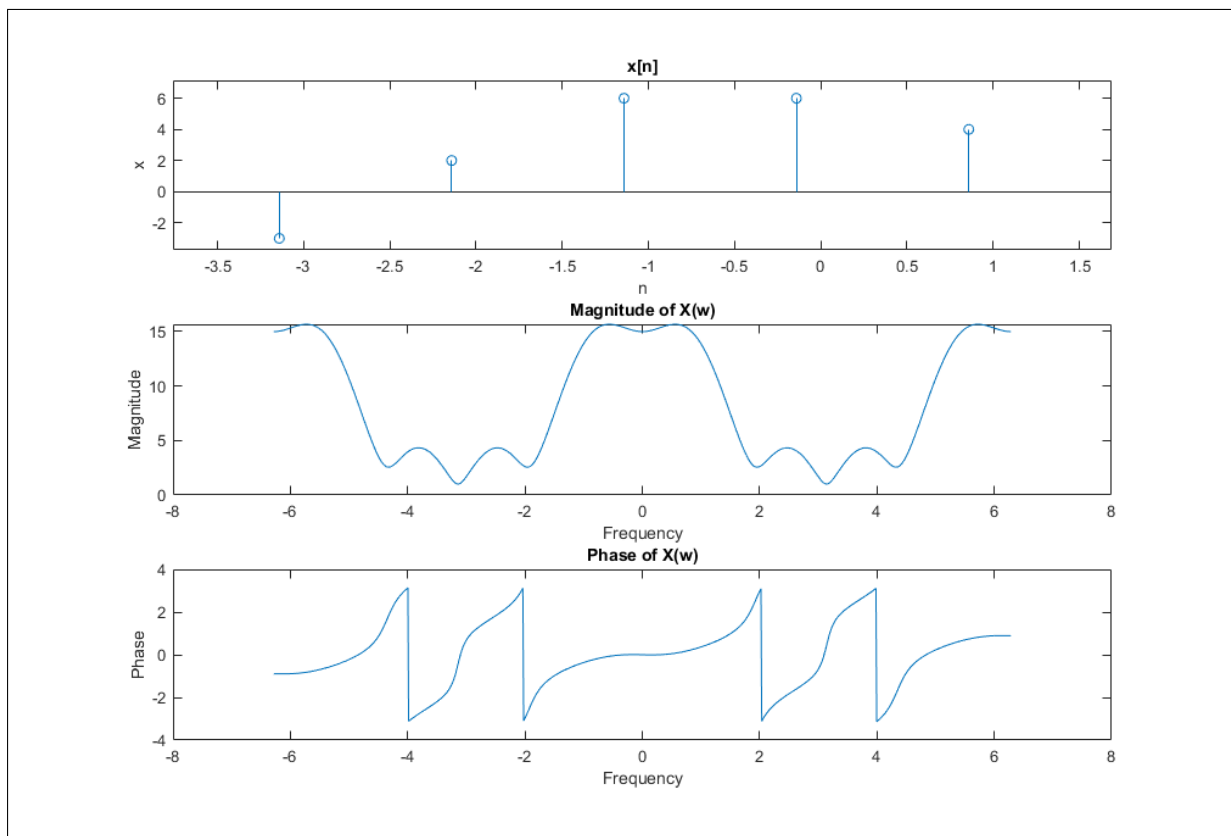


Figure 2: DTFT of signal $x[n + \pi]$

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:

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

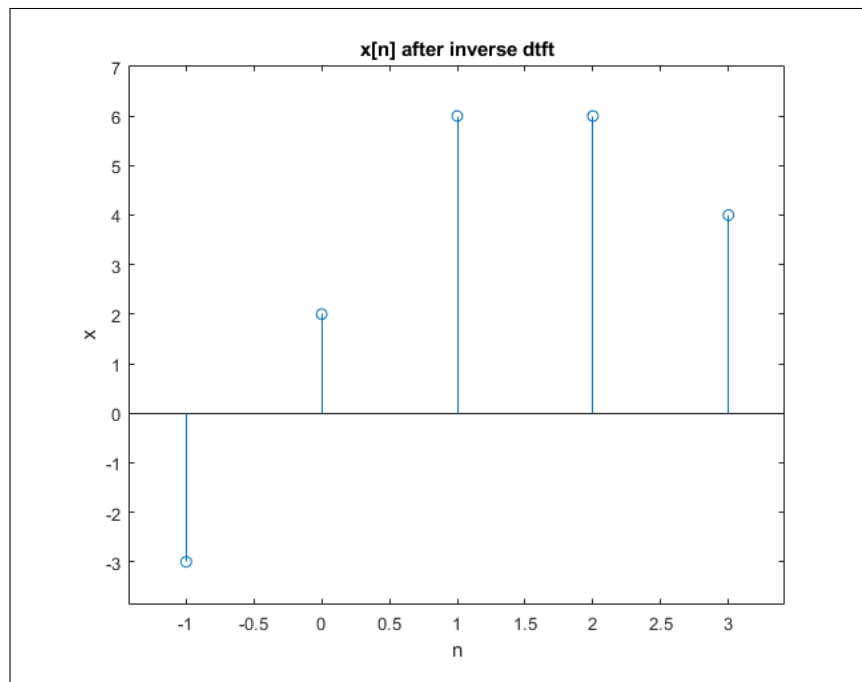


Figure 3: Inverse DTFT of $X(e^{j\omega})$

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 if it is solely real.

3 Filters

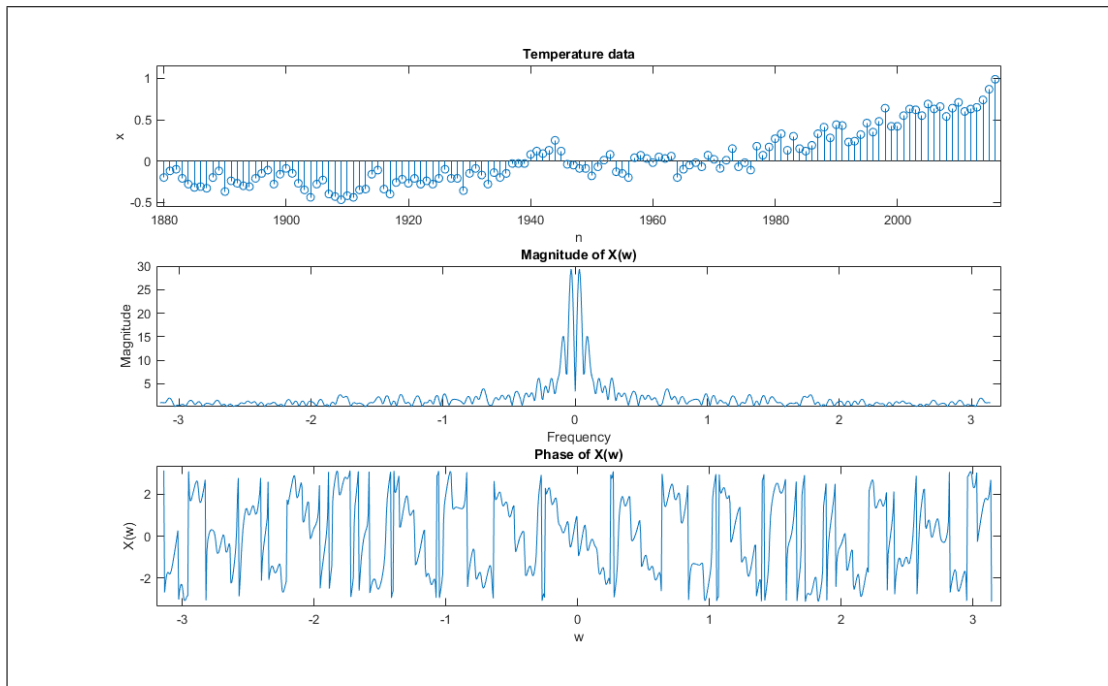


Figure 4: DTFT of temperature data

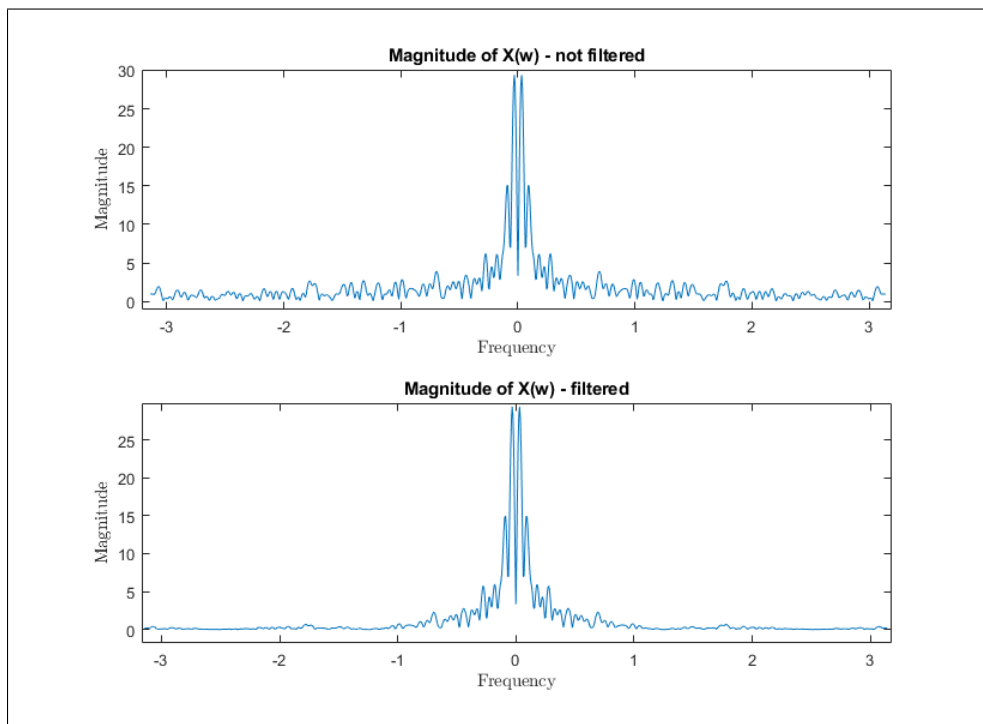


Figure 5: Comparison of un-filtered and filtered data

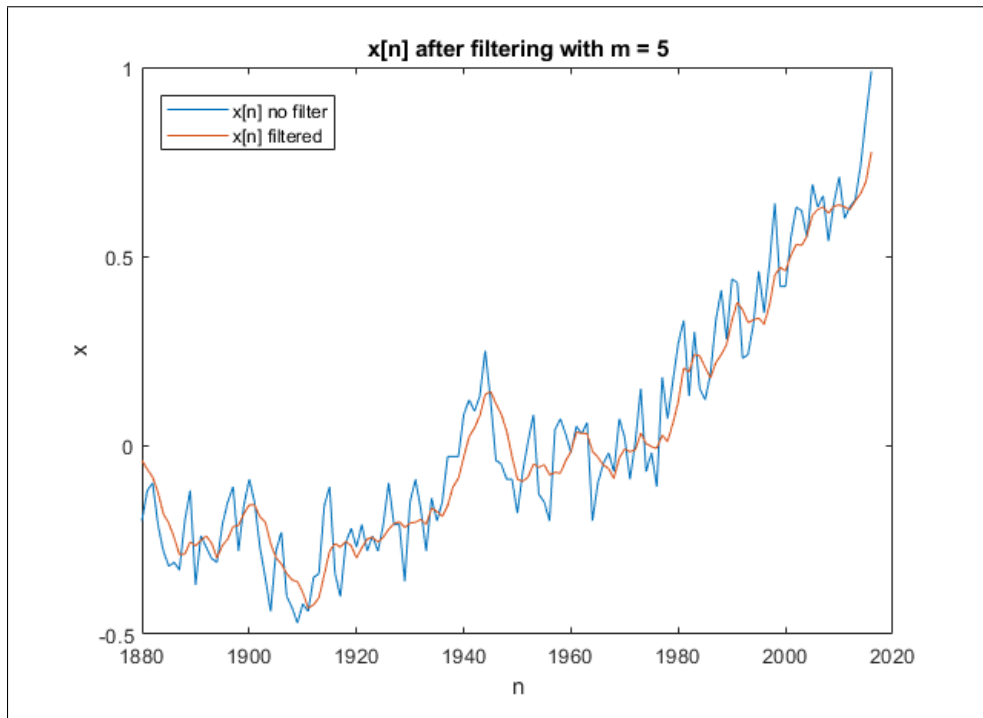


Figure 6: DTFT of temperature data

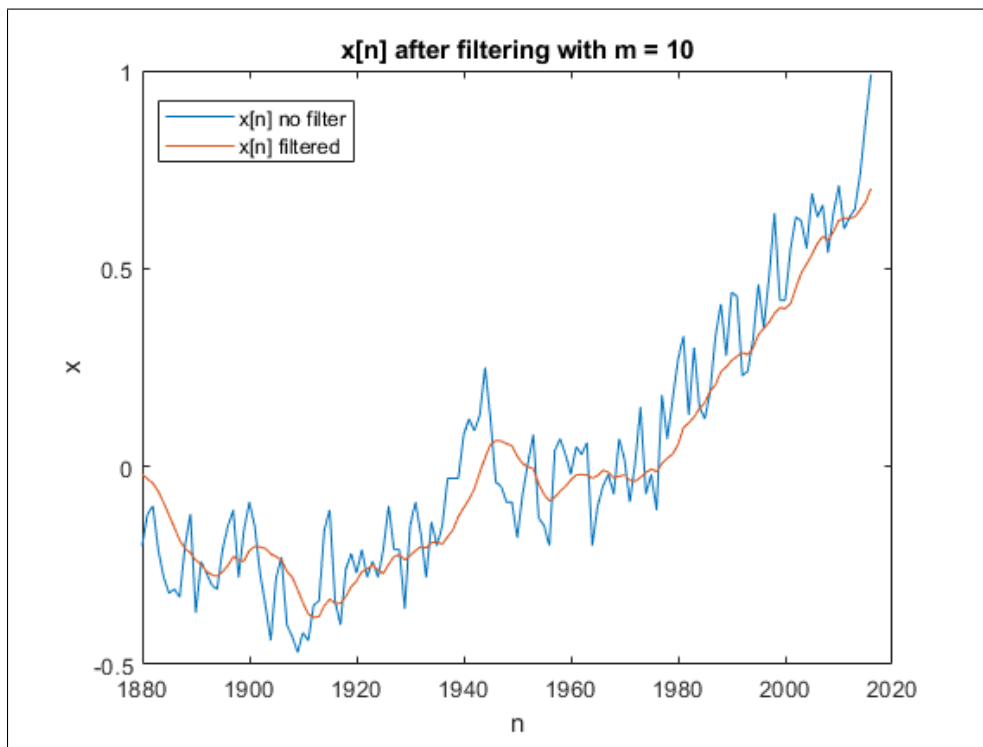


Figure 7: Comparison of un-filtered and filtered data

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

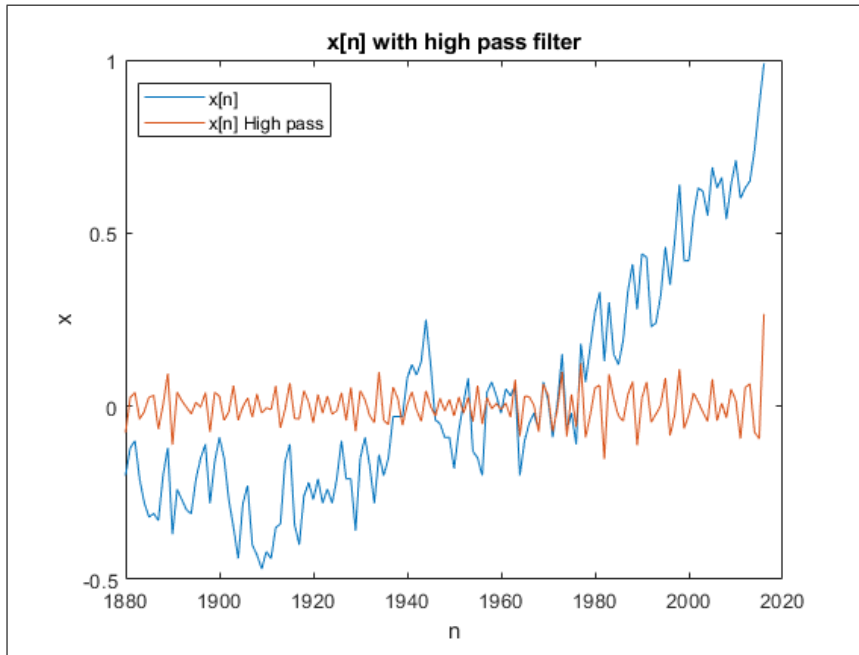


Figure 8: DTFT of temperature data

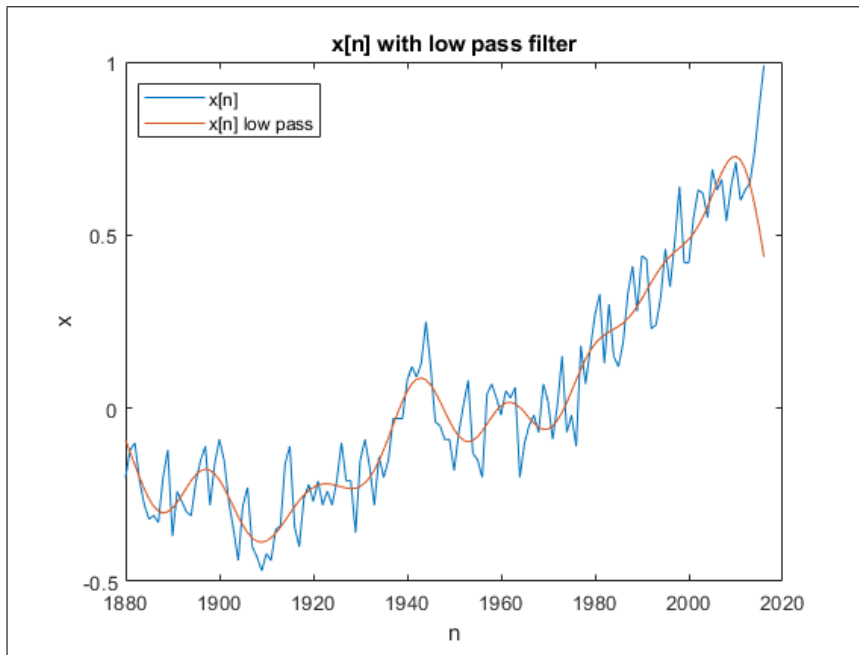


Figure 9: Comparison of un-filtered and filtered data

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

```
1 %% 1: dtft
2 clc , clear
3
4 M = 1000;
5 k = (0 : M)';
6 omega_0 = -2*pi;
7 omega_M = 2*pi;
8 omega = omega_0 + (omega_M-omega_0)*(k/M);
9
10 x = [-3,2,6,6,4]';
11 n = (0:length(x)-1)'-pi;
12 X = dtft( x, n, omega);
13
14 subplot(3, 1, 1);
15 stem(n, x);
16 title('x[n]')
17 xlabel('n')
18 ylabel('x')
19
20 subplot(3,1,2);
21 plot(omega, abs(X));
22 title("Magnitude of X(w)")
23 xlabel('Frequency')
24 ylabel('Magnitude')
25
26 subplot(3,1,3);
27 plot(omega, angle(X));
28 title("Phase of X(w)")
29 xlabel('Frequency')
30 ylabel('Phase')
31
32 %% 2: Inverse dtft
33 clc
34
35 x1 = invdtft(X, n, omega)
36 stem(n, x1);
37 title('x[n] after inverse dtft')
38 xlabel('n')
39 ylabel('x')
40
41
42 %% 3a: dtft of Temperature data
43 clc , clear
44
45 data = importdata('Temperature.txt');
46 n = data(:, 1);
47 x = data(:, 2);
48
49 omega_0 = -pi;
50 omega_M = pi;
51 M = 1000;
52 k = (0 : M)';
```

```

53 omega = omega_0 + (omega_M-omega_0)*(k/M);
54
55 X = dtft( x, n, omega);
56
57 subplot(3, 1, 1);
58 stem(n, x);
59 title("Temperature data")
60 xlabel('n')
61 ylabel('x')
62
63 subplot(3,1,2);
64 plot(omega,abs(X));
65 title("Magnitude of X(w)")
66 xlabel('Frequency')
67 ylabel('Magnitude')
68
69 subplot(3,1,3);
70 plot(omega,angle(X));
71 title("Phase of X(w)")
72 xlabel('w')
73 ylabel('X(w)')
74
75
76
77 %% 3: moving average filter
78
79 m = 10;
80 Hav = ((1/m).*exp(-1j.*omega.*((m-1)./2)).*((sin(omega.*m./2))./(sin(
    omega./2)))));
81 Hav((M/2)+1) = 1;
82
83 Y = Hav.*X;
84 y = invdtft(Y, n, omega);
85
86 figure(1)
87 plot(n,x, n,y)
88 title("x[n] after filtering with m = 10")
89 xlabel('n')
90 ylabel('x')
91 legend('x[n] no filter', 'x[n] filtered');
92
93 figure(2)
94
95 subplot(2,1,1);
96 plot(omega,abs(X));
97 title("Magnitude of X(w) - not filtered")
98 xlabel('Frequency','interpreter','latex')
99 ylabel('Magnitude','interpreter','latex')
100
101 subplot(2,1,2);
102 plot(omega,abs(Y));
103 title("Magnitude of X(w) - filtered")
104 xlabel('Frequency','interpreter','latex')
105 ylabel('Magnitude','interpreter','latex')
106

```



```

107
108 %% extra
109 clc
110
111 wc = pi/2;
112 high_filter = (abs(omega) > wc);
113
114 Y = high_filter.*X;
115 y = invdtft(Y, n, omega);
116
117 figure(1)
118 plot(n,x, n,y)
119 title("x[n] with high pass filter")
120 xlabel('n')
121 ylabel('x')
122 legend('x[n]', 'x[n] high pass');
123
124 wc = pi/8;
125 low_filter = (abs(omega) < wc);
126 Y = low_filter.*X;
127 y = invdtft(Y, n, omega);
128
129 figure(2)
130 plot(n,x, n,y)
131 title("x[n] with low pass filter")
132 xlabel('n')
133 ylabel('x')
134 legend('x[n]', 'x[n] low pass');

```