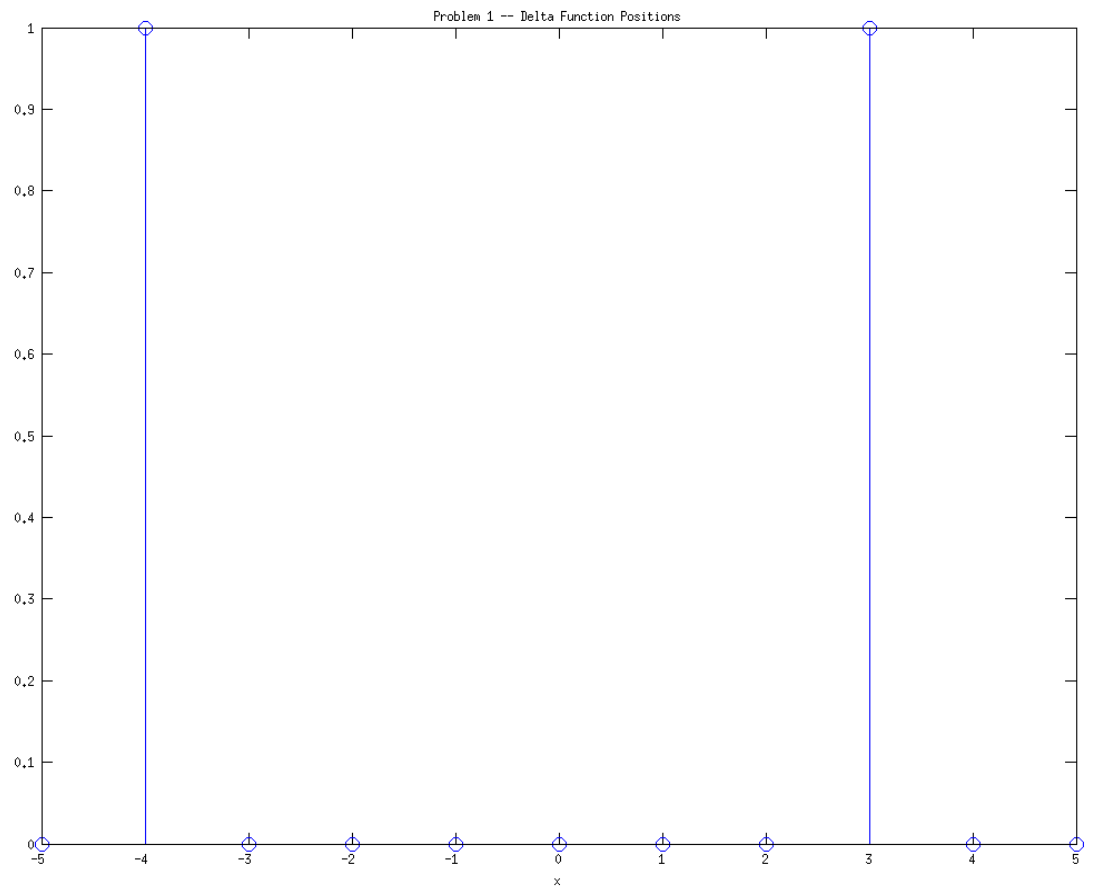


Advanced Digital Signal Processing Coursework
Candidate Number : 137377

December 9, 2015

1 Problem 1a

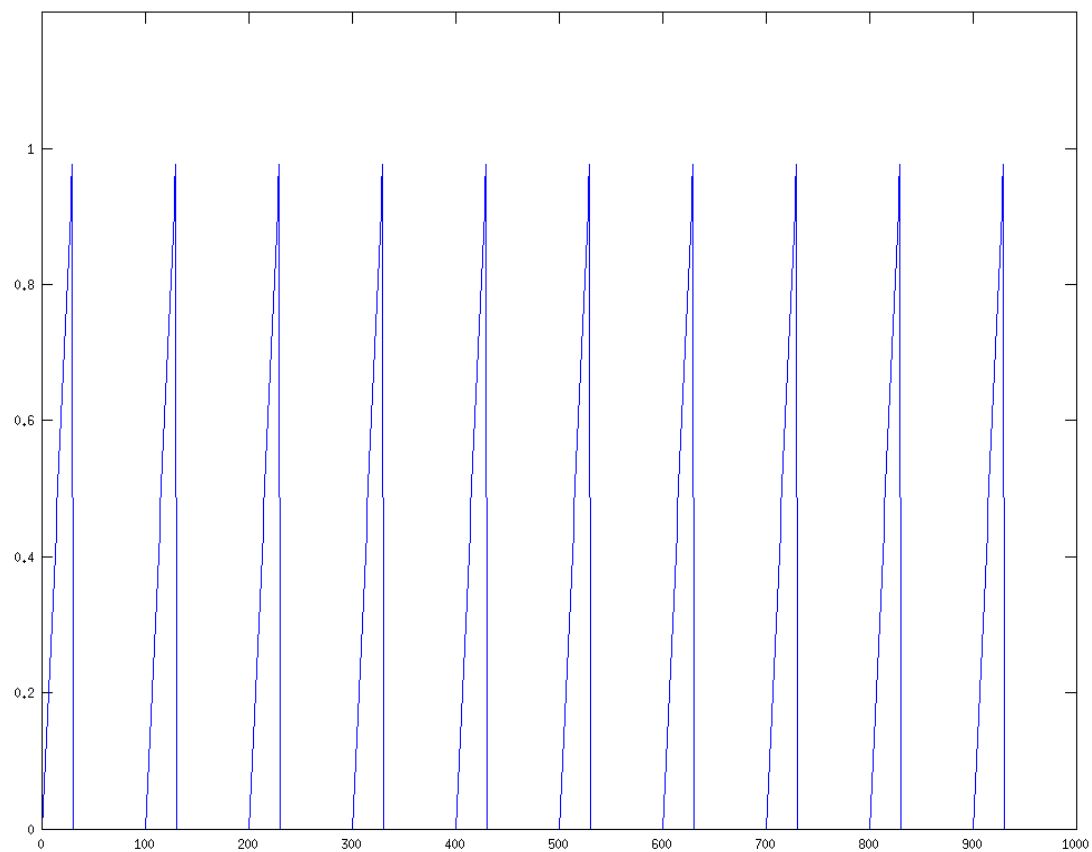
```
1 % Problem # 1a Soln.:
2 %-----Delta Function Plotting-----
3 %-----
4 %-----Candidate No. 137377 -----
5
6 x=-5:5;
7 m = zeros(1,size(x,2));
8 % Delta function at x=-4
9 m(2) = 1;
10 % Delta function at x= 3
11 m(9) = 1;
12
13 h = figure;
14 circshift(m,-4,1);
15 stem(x,m)
16 xlabel('x')
17 title('Problem 1 — Delta Function Positions')
```

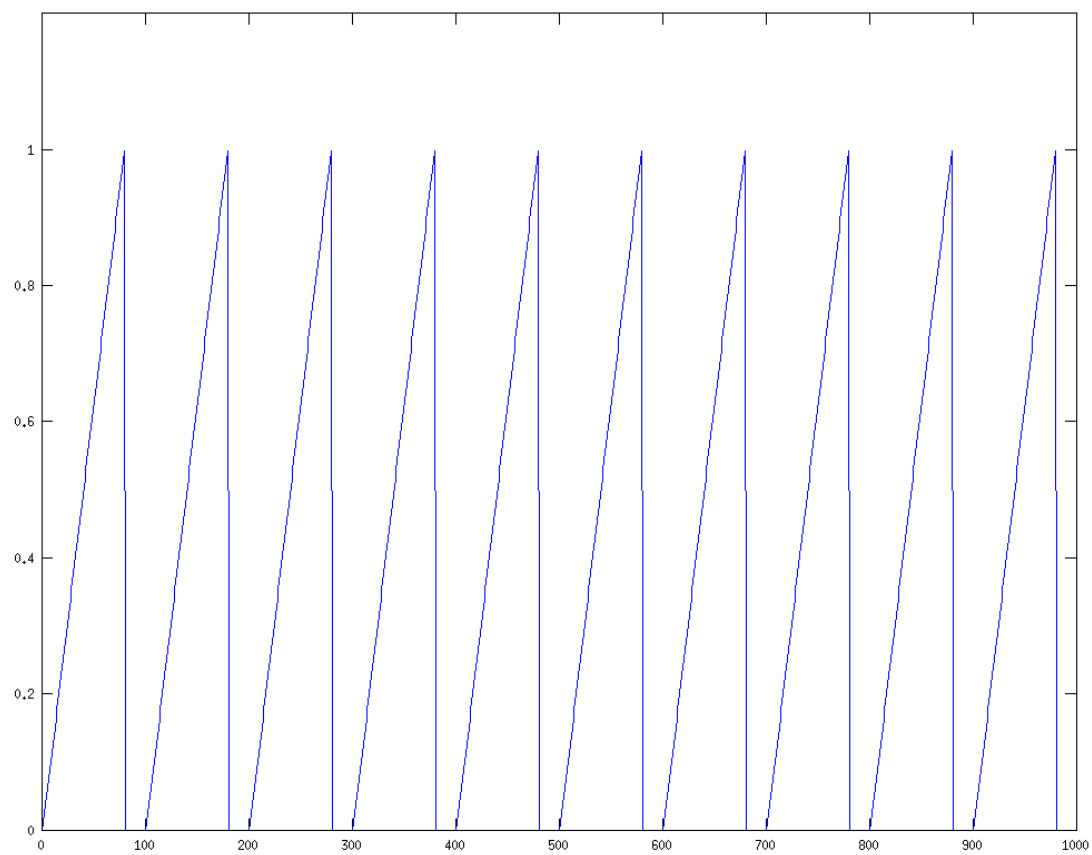


2 Problem 1b

```
1 % Problem # 1b (i & ii) Soln.:
2 %-----Sawtooth Function Plotting-----
3 %-----
4 %-----Candidate No. 137377 -----
5 %Generate a sawtooth waveform with a chosen duty cycle:
6 clear all; close all;
7
8 %Get the duty cycle percentage. Should be less than 100%.
9 duty = mod(ceil(input('Enter desired Duty Cycle (%) : '))
    ,100);
10
11 %Generate the sawtooth shape
12 f = @(x) [x/duty.*(0<=x & x<=duty)];
13 x = linspace(0,100);
14 y = f(x);
15
16 %Number of times to repeat sawtooth in plot
17 repetitions = 10;
18 yy = y;
19 for i=1:repetitions;
20 yy = cat(2,yy,y);
21 end
22
23 % Plot
24 h = figure(1);
25 plot(1:length(yy),yy)
26 ylim([0 1.2]);
27 xlim([0 repetitions*100]);
```

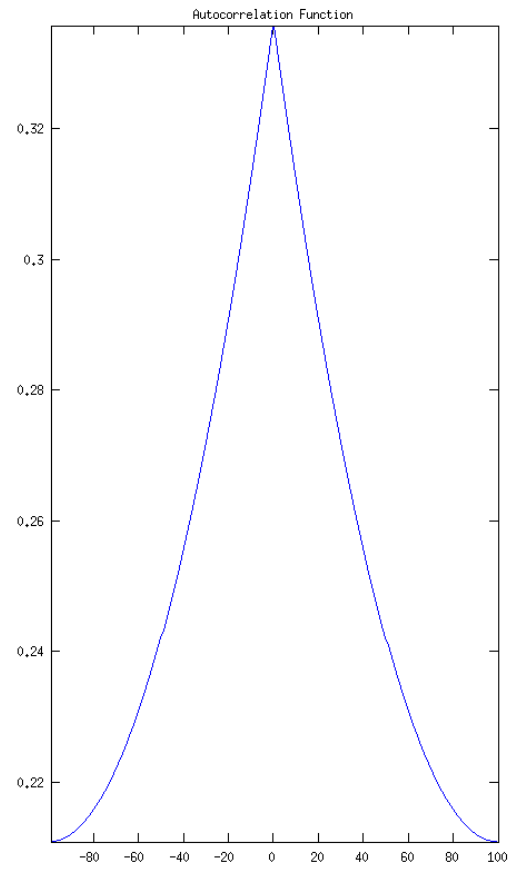
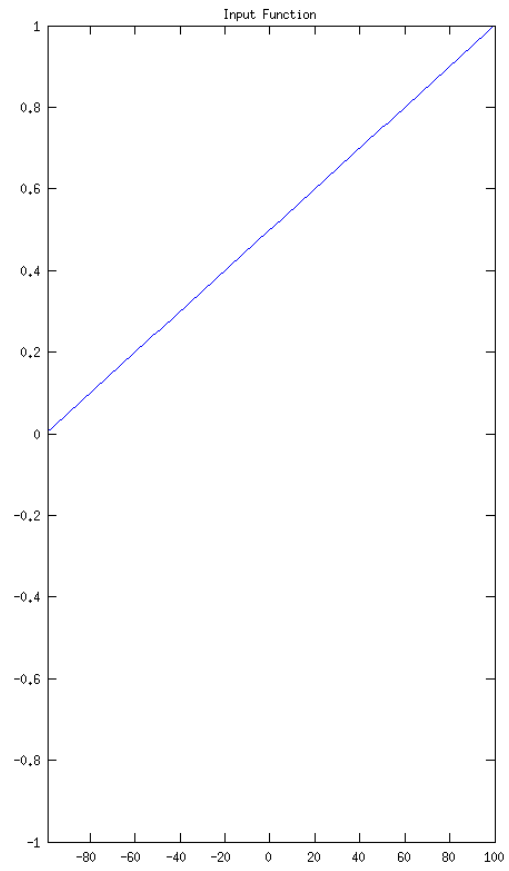
Showing output for 30% and 80% duty cycles respectively.





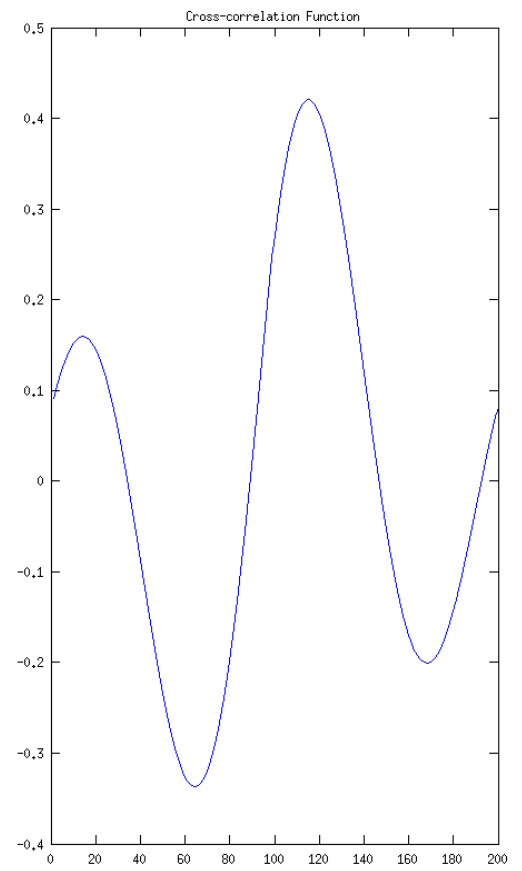
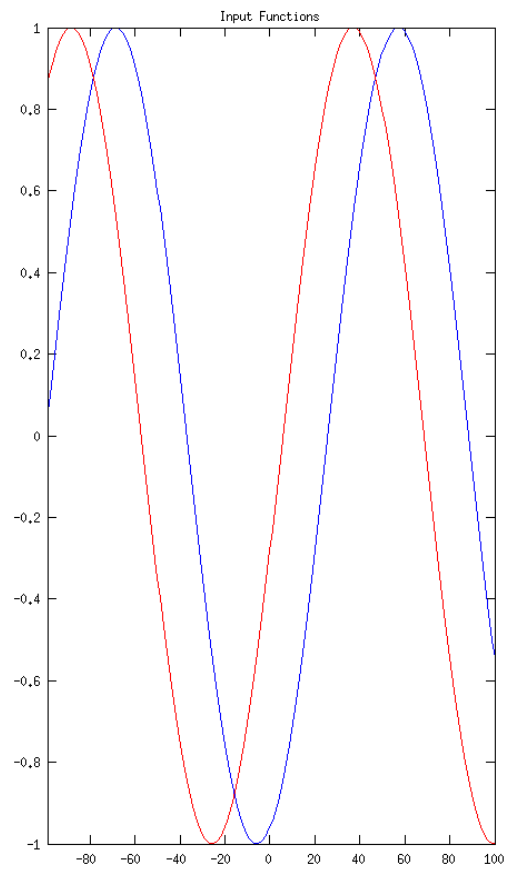
3 Problem 2(i)

```
1 % Problem # 2(i) Soln.:
2 %-----Autocorrelation-----
3 %-----
4 %-----Candidate No. 137377 -----
5 %Auto-correlate a sequence with itself
6 clear all; close all;
7
8 N = 200;
9 % Simple Line Input
10 x = (1:200)/N;
11
12
13 % Initialize array to hold the autocorrelation data
14 psi = x*0;
15
16 for i = 1:N
17     xshift = circshift(x,[0,(i + N/2)]);
18     psi(i) = N^-1 * sum(x .* xshift);
19 end
20
21 h= figure;
22 subplot(1,2,1)
23 plot(-floor(N/2)+1:floor(N/2),x);
24 title('Input Function');
25 axis([-floor(N/2)+1 floor(N/2) -1 1]);
26 subplot(1,2,2)
27 plot(-floor(N/2)+1:floor(N/2),psi);
28 title('Autocorrelation Function');
29 axis([-floor(N/2)+1 floor(N/2) min(psi) max(psi)])
```



4 Problem 2(ii)

```
1 % Problem # 2(ii) Soln.:
2 %-----Cross Correlation-----
3 %-----
4 %-----Candidate No. 137377 -----
5 % Cross correlation of two sequences.
6 clear all; close all;
7
8 N = 200;
9 % sin-shaped signal — slowly varying
10 x(1:N) = sin((1:N) / 20);
11 y(1:N) = sin(((1:N)+20) / 20);
12
13 % Initialize array to hold the correlation data
14 psi = x*0;
15
16 % Apply shift to cross-correlate
17 for i = 1:N
18     yshift = circshift(y,[0,(i + N/2)]);
19     psi(i) = N^-1 * sum(x .* yshift);
20 end
21
22 h = figure
23 subplot(1,2,1)
24 plot(-floor(N/2)+1:floor(N/2),x,'b');
25 title('Input Functions');
26 axis([-floor(N/2)+1 floor(N/2) -1 1]);
27 hold on;
28 plot(-floor(N/2)+1:floor(N/2),y,'r');
29 subplot(1,2,2)
30 plot(1:size(psi,2),psi);
31 title('Cross-correlation Function');
```



5 Problem 3

```

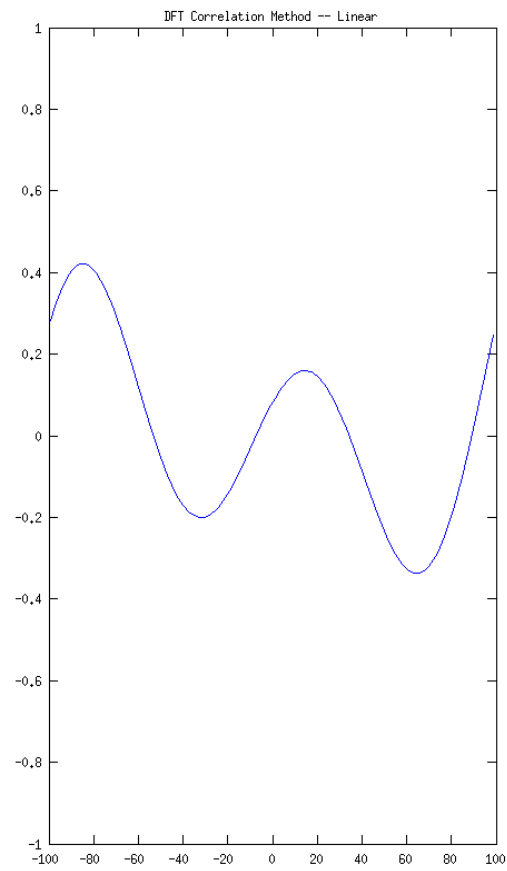
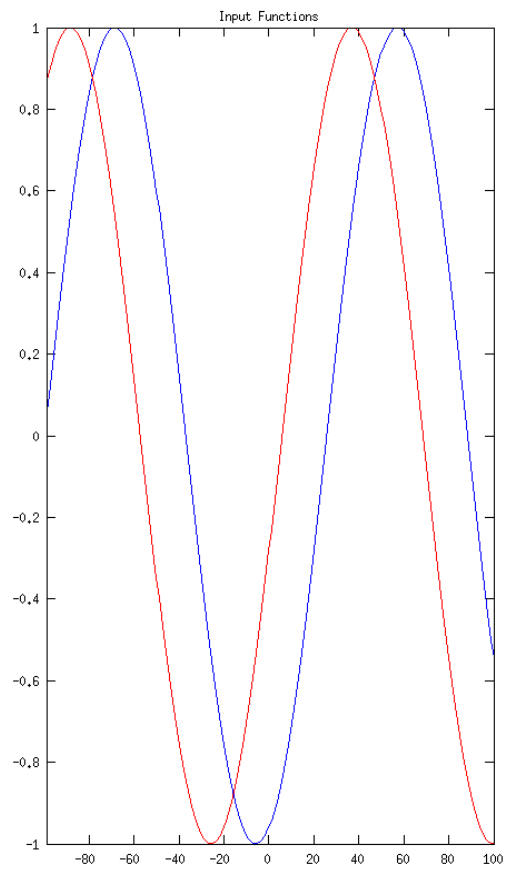
1 % Problem # 3 Soln.:
2 %-----Comparison of Correlation Methods-----
3 %-----
4 %-----Candidate No. 137377 -----
5 %
6 % Graph 1 shows the input functions , Graph 2 shows the
   linear correlation , Graph 3 shows the circular
   correlation , Graph 4 shows the summed version of Graph
   3 to show that it is the same as Graph 2.
7 % Ultimately , the different methods produce the same
   output in Graphs 2 and
8 % 4. However, the zero-padding circular method produces
   an output that is
9 % periodic and must be summed to produce the final result
   .
10
11 clear all; close all;
12
13 N = 200;
14 % sin-shaped signal — slowly varying
15 x(1:N) = sin((1:N)/20);
16 x_f = fft(fftshift(x));
17 y(1:N) = sin(((1:N)+20) / 20);
18 y_f = fft(fftshift(y));
19
20 % Correlation without Padding
21 % Do the multiplication in frequency space
22 psi_f = x_f .* conj(y_f);
23 psi_1 = (ifft(psi_f))/N;
24
25 % Correlation with Zero Padding
26 % Pad arrays to size N >= N_1 + N_2 - 1
27 pad_size = size(x,2) + size(y,2) - 1;
28 x_padded = zeros(1,pad_size); x_padded(size(x,2):end) = x
   (1:end);
29 y_padded = zeros(1,pad_size); y_padded(1:size(y,2)) = y
   (1:end);
30 x_padded_f = fft(fftshift(x_padded));
31 y_padded_f = fft(fftshift(y_padded));
32
33 y_save = x_padded_f .* conj(y_padded_f);
34 y_save = ifftshift(ifft(y_save));
35

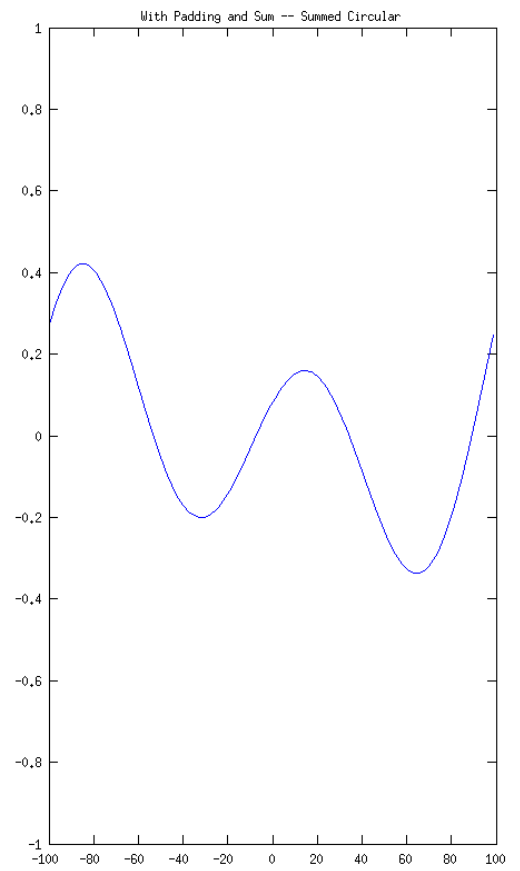
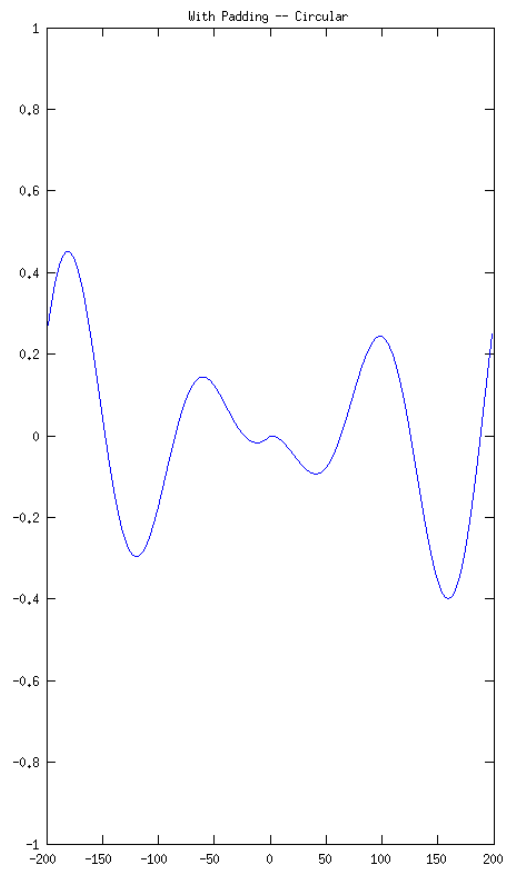
```

```

36 psi_2 = y_save / N;
37 psi_2_sum = (y_save(1:N) + y_save(N:end)) / N;
38
39
40 h = figure
41 subplot(1,4,1)
42 plot(-floor(N/2)+1:floor(N/2),x,'b');
43 title('Input Functions');
44 axis([-floor(N/2)+1 floor(N/2) -1 1]);
45 hold on;
46 plot(-floor(N/2)+1:floor(N/2),y,'r');
47 subplot(1,4,2)
48 plot(-N/2:N/2-1,psi_1);
49 title('DFT Correlation Method — Linear');
50 axis([-N/2 N/2 -1 1])
51 subplot(1,4,3)
52 plot(-N+1:N-1,psi_2); hold on;
53 title('With Padding — Circular')
54 axis([-N N -1 1])
55 subplot(1,4,4)
56 plot(-N/2:N/2-1,psi_2_sum);
57 title('With Padding and Sum — Summed Circular')
58 axis([-N/2 N/2 -1 1])

```





6 Problem 4

```

1 % Problem # 4 Soln.:
2 %-----IIR Filter Design-----
3 %-----
4 %-----Candidate No. 137377 -----
5 % Computes the frequency response through the bilinear
   transform method of
6 % a high-pass IIR filter.
7 % See report for the derivation of G.
8
9 clear all; close all;
10
11 % point_1_f = input('Enter desired -3dB cut-off frequency
   : ');
12 % point_2 = input('Enter next attenuation point (dB): ');
13 % point_2_f = input('Enter desired frequency at the next
   attenuation point : ');
14 % sample_rate = input('Enter sampling rate : ');
15
16 % Inputs used for graph in report
17 point_1_f = 1000;
18 point_2 = 10;
19 point_2_f = 350;
20 sample_rate = 5000;
21
22 T = 1/sample_rate;
23 w = 2*pi*(0:sample_rate/2);
24 w_ac = (2/T) * tan(2*pi * point_1_f * T / 2);
25 w_att = (2/T) * tan(2*pi * point_2_f * T / 2);
26
27 filter_order = round(log10(10^(point_2/10) - 1) / (2*
   log10(w_ac/w_att)));
28 disp(['Filter has order: ', num2str(filter_order)]);
29
30 % Generate range of frequencies based on sample rate
31 z = exp(1i*w*T);
32 % Apply substitution for s for the type of filter, and
   the bilinear
33 % transform
34 s = ((2/T) .* ((z-1)./(z+1))).^filter_order;
35 G = s ./ (s + (-1 ^ filter_order) * (w_ac)^filter_order);
36
37 % High-pass filter:
38

```

```

39
40 % Plot up until operating range of filter at z=-1
41 h = figure;
42 plot(w,abs(G).^2);
43 xlabel('Frequency');
44 ylabel('|G(f)|^2');
45 title('Frequency Response of Filter')

```


The following were derived for use in the program:

$$\text{Substitution for } s \text{ in the high-pass filter case : } s = \frac{\omega_a}{s} \quad (1)$$

$$G(s) = \frac{1}{1 + (-1)^n \left(\frac{\omega_{ac}^n}{s^n} \right)} \quad (2)$$

$$G(s) = \frac{s^n}{s^n + (-1)^n (\omega_{ac}^n)} \quad (3)$$

$$\text{Substitution for bilinear transform : } s = \frac{2}{T} \frac{z-1}{z+1} \quad (4)$$

$$G(z) = \frac{\left(\frac{2}{T} \frac{z-1}{z+1} \right)^n}{\left(\left(\frac{2}{T} \frac{z-1}{z+1} \right)^n + (-1)^n (\omega_{ac}^n) \right)} \quad (5)$$

$$(6)$$

Graph of the frequency response of the filter. With a -3dB cut-off frequency of 1 kHz, attenuation of 10 dB at 350 Hz, and a sampling frequency of 5 KHz.

