ELE632 Lab 3 Report

Table of Contents

Authors: Mai Abdelhameed & Omar Ahmad

A. Discrete-Time Fourier Series

A.1

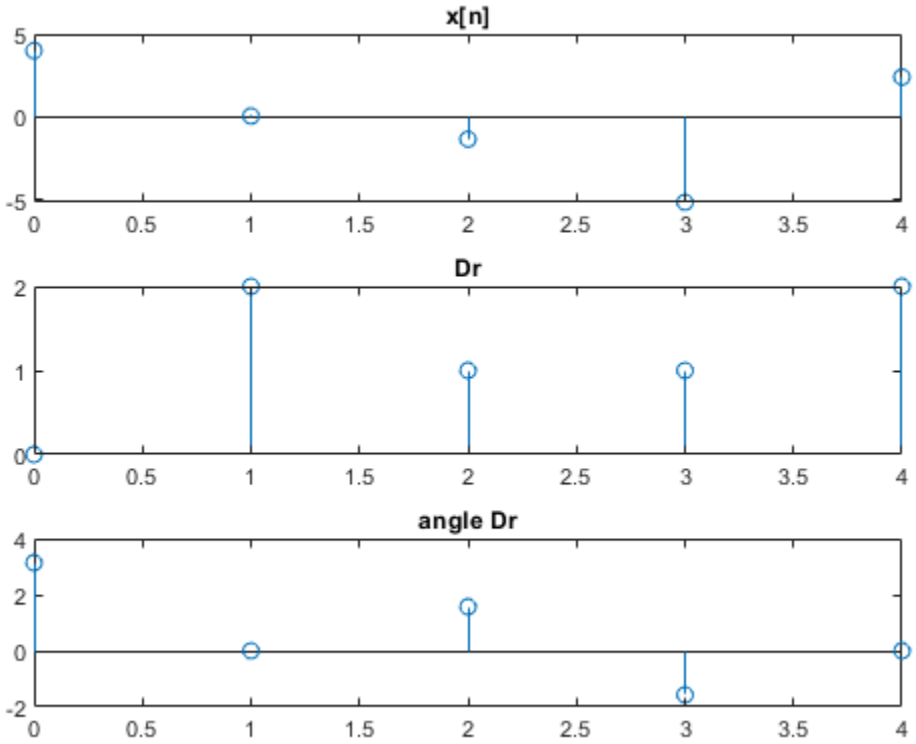
```
%x[n] = 4cos((12/5) pi n) + 2sin((16/5) pi n)
disp('fundamental frequency w0 = 0.4pi')
disp('fundamental period N0 = 5')
```

```
fundamental frequency w0 = 0.4pi
fundamental period N0 = 5
```

A.2

```
N0 = 5;
w0= 0.4*pi;

n =(0:4);   %0:N0-1

x = @(n) (4.*cos(2.4.*pi.*n) + 2.*sin(3.2.*pi.*n));

for r = 0:4
    xr(r+1) = sum(x(n).*exp(-j*r*(0.4*pi)*n))/5;
end

r = n;

subplot(3,1,1);
a = (0:4);
b = x(a);
stem(a, b);
```

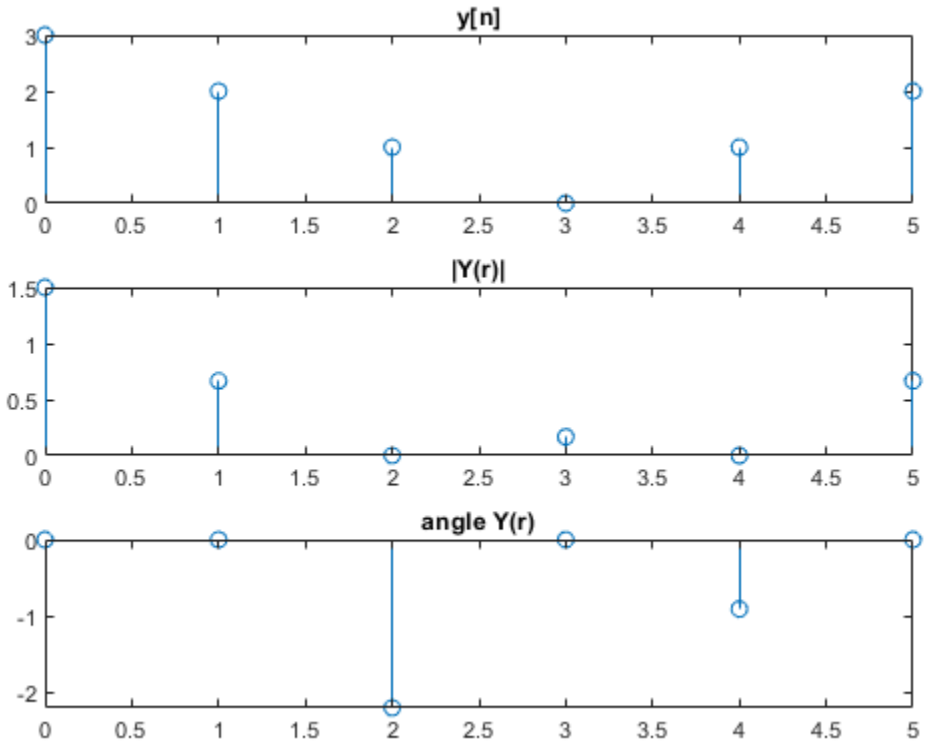
```
title('x[n]');

subplot (3,1,2);
stem(r, abs(xr));
title ('Dr');

subplot  (3,1,3);
stem(r, angle(xr));
title ('angle Dr');
```



A.3

```
N0 = 6;
w0= pi/3;

n =(0:5);   %0:N0-1
yn = [3 2 1 0 1 2];

for r = 0:5
    yr(r+1) = sum(yn.*exp(-j*r*(pi/3)*n))/6;
end

r = n;

subplot(3,1,1);
a = (0:5);
```

```
stem(a, yn);
title('y[n]');

subplot (3,1,2);
stem(r, abs(yr));
title ('|Y(r)|');

subplot  (3,1,3);
stem(r, angle(yr));
title ('angle Y(r)');
```



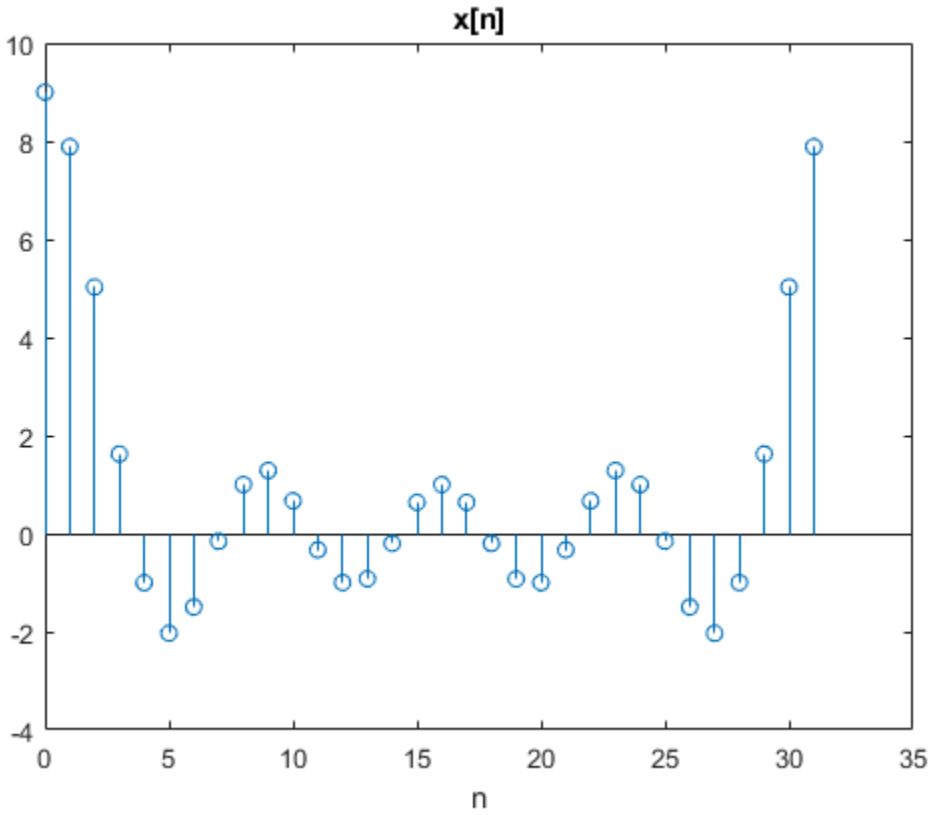
B. Inverse DTFS and time shifting property

B.1

```
N0 = 32;
n = (0:31);
w0 = pi/16;
Xr = [ones(1,5) zeros(1,23) ones(1,4)];

for r = 0:31
    xn(r+1) = sum(Xr.*exp(j*n*(pi/16)*r));
end
r=n;
figure;
```

```
stem(n,xn)
xlabel('n');
title('x[n]');
```

Warning: Using only the real component of complex data.

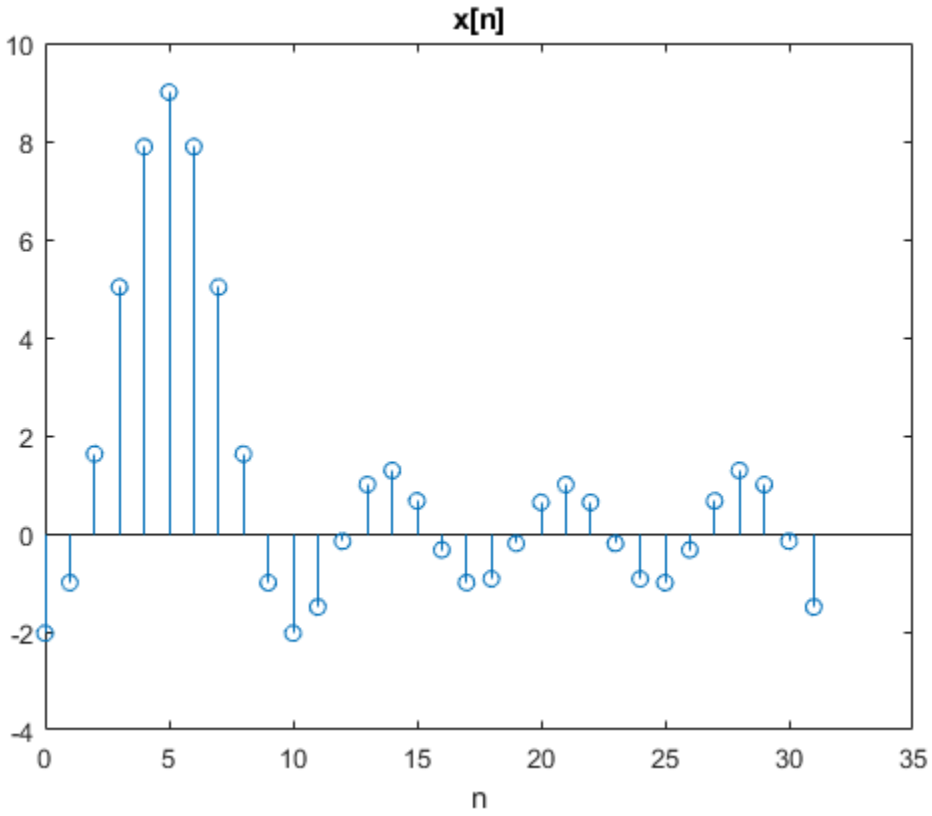


B.2

```
N0 = 32;
n = (0:31);
w0 = pi/16;
Xr = [ones(1,5) zeros(1,23) ones(1,4)].*exp(-j*5*(pi/16)*n);


for r = 0:31
    xn(r+1) = sum(Xr.*exp(j*n*(pi/16)*r));
end
r = n;
figure;
stem(r, xn);
xlabel('n');
title('x[n]');

disp('x[n] of B.2 is a shifted version of the x[n] of B.1 by 5.');
```

Warning: Using only the real component of complex data.

x[n] of B.2 is a shifted version of the x[n] of B.1 by 5.

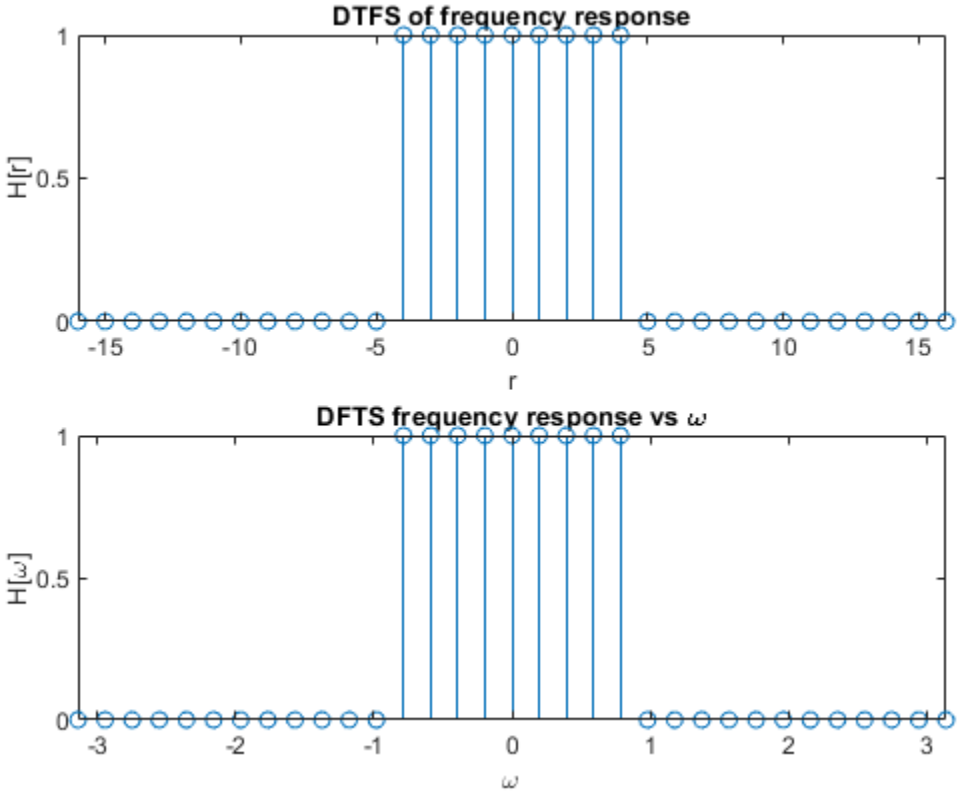


C. System Response

C.1

```
clear;

u = @(n) (n >= 0) * 1.0 .* (mod(n,1)==0);
H = @(n) u(n+4)-u(n-5);
N_0 = 32;
n = (-16:16);
omega = (2*pi/N_0);

subplot(2, 1, 1);
stem(n,H(n));
xlabel('r');
ylabel('H[r]');
axis([-16 16 0 1]);
title('DTFS of frequency response');

subplot(2, 1, 2);
stem(n.*omega,H(n));
xlabel('\omega');
ylabel('H[\omega]');
```

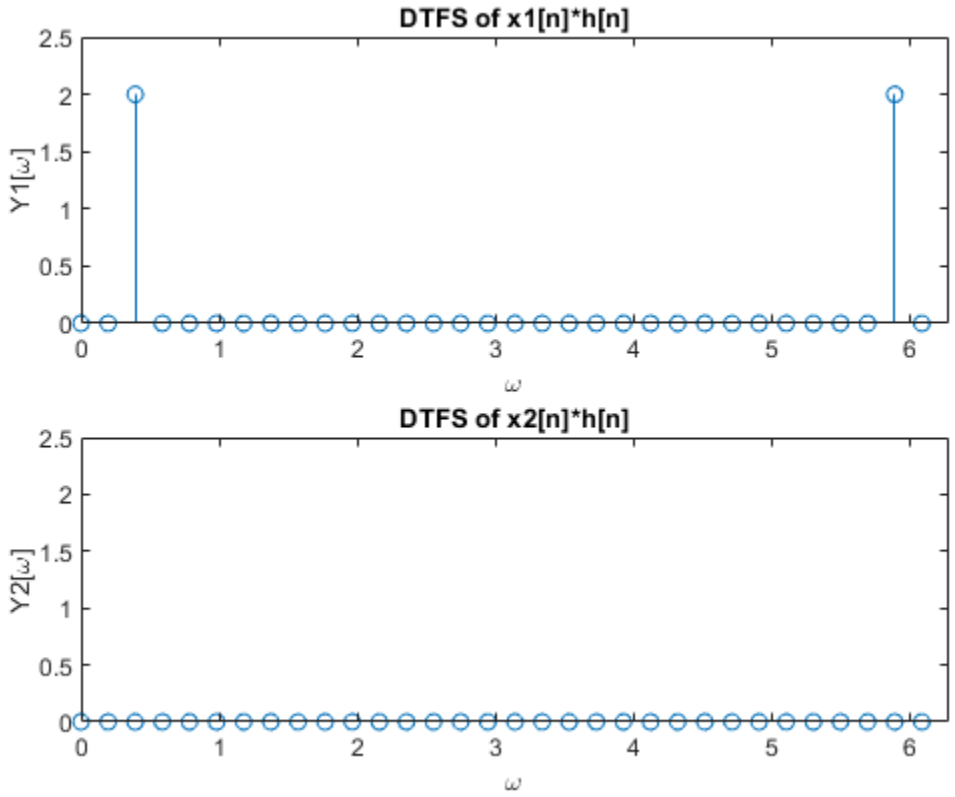
```
axis([-pi pi 0 1]);
title ('DFTS frequency response vs \omega');
```



C.2-3

```
clear;

N_0 = 32;
omega = 2*pi/N_0;
n = (0:N_0-1);
H1 = [ones(1,5) zeros(1,23) ones(1,4)];
x1 = @(n) 4.0*cos(pi*n/8);
x2 = @(n) 4.0*cos(pi*n/2);

for r = (0:N_0-1)
    X1(r+1) = sum(x1(n).*exp(-j.*r.*(omega).*n))/N_0;
    X2(r+1) = sum(x2(n).*exp(-j.*r.*(omega).*n))/N_0;
end

figure;

subplot(2,1,1);
stem(n.*omega, H1.*abs(X1));
axis([0 2*pi 0 2.5]);
title('DTFS of x1[n]*h[n]');
xlabel('\omega');
```

```
ylabel('Y1[\omega]');

subplot(2,1,2);
stem(n.*omega, H1.*abs(X2));
axis([0 2*pi 0 2.5]);
title('DTFS of x2[n]*h[n]');
xlabel('\omega');
ylabel('Y2[\omega]');
```



C.4

```
disp('H[r] is a low-pass filter, which lets through signals between -
pi/4');
disp('to pi/4. Since both input signals are sinusoidal, they produce
 a');
disp('delta function at the location of \omega. x1 produces it at
 pi/8');
disp('which is within the filter, which is why it was displayed, while
 the');
disp('second signal was blocked out due to it not being within the
 range.')
```

```
H[r] is a low-pass filter, which lets through signals between -pi/4
to pi/4. Since both input signals are sinusoidal, they produce a
delta function at the location of \omega. x1 produces it at pi/8
which is within the filter, which is why it was displayed, while the
```

```
second signal was blocked out due to it not being within the range.
```

Published with MATLAB® R2018b