
```
%Ramaseshan Parthasarathy
%RUID: 157006637
%12/05/16
```

PROBLEM 1: AC/DC HALF-WAVE RECTIFIER & CONVERTER

```
%An AC-to-DC converter or 'rectifier' takes an AC input x(t) and
converts
%it to a DC outout y(t) while passing it through a diode followed by
LPF.
%This is achieved in matlab by using M-term approximation to fourier
%series.

%PART A
w0 = 2*pi;
T = (2*pi)/w0;
c0 = 1/pi;
c1 = 1/4i;
M = 10; tau = T;

H = @(s) 1/(1 + s*tau);
P = @(s) w0 * (1 + exp(-s*T/2))/(s^2 + w0^2)
c = @(k) (1/T) * P(j*k*w0)

ctr = 1;
for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;
figure; plot(t, fM, 'b', t, yMsteady, 'r');
title('half-wave rectifier, M = 10, \tau = 1T');

M = 30; tau = T;

H = @(s) 1/(1 + s*tau);

ctr = 1;
```

```

for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;
figure; plot(t, fM, 'r', t, yMsteady, 'r');
title('half-wave rectifier, M = 30, \tau = 1T');

M = 10; tau = 5*T;
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;
figure; plot(t, fM, 'b', t, yMsteady, 'r');
title('half-wave rectifier, M = 10, \tau = 5T');

M = 30; tau = 5*T;
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end

```

```

        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;
figure; plot(t, fM, 'b', t, yMsteady, 'r');
title('half-wave rectifier, M = 30, \tau = 5T');

M = 10; tau = 10*T;
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;
figure; plot(t, fM, 'b', t, yMsteady, 'r');
title('half-wave rectifier, M = 10, \tau = 10T');

M = 30; tau = 10*T;
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:3*T
    fM(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM(ctr) = fM(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    ctr = ctr + 1;
end

t = 0:0.01:3*T;

```

```

figure; plot(t, fM, 'b', t, yMsteady, 'r');
title('half-wave rectifier, M = 30, \tau = 10T');

%PART B
M = 30; tau = 5*T;

A = -(tau^(-1) * P(-1*tau^(-1)))/(exp(T/tau) - 1);
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:24*T
    fM1(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM1(ctr) = fM1(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    yM1(ctr) = A*exp(-1*t/tau) + yMsteady(ctr);
    ctr = ctr + 1;
end

t = 0:0.01:24*T;
figure; plot(t, fM1, 'b', t, yMsteady, 'r', t, yM1, 'k');
title('half-wave rectifier, M = 30, \tau = 5T');

M = 30; tau = 10*T;

A = -(tau^(-1) * P(-1*tau^(-1)))/(exp(T/tau) - 1);
H = @(s) 1/(1 + s*tau);

ctr = 1;
for t = 0:0.01:24*T
    fM2(ctr) = c0;
    yMsteady(ctr) = c0*H(0);
    for k = 1:M
        ck = c(k);
        if k == 1
            ck = c1;
        end
        fM2(ctr) = fM2(ctr) + 2 * real(ck * exp(1i*k*w0*t));
        yMsteady(ctr) = yMsteady(ctr) + 2 * real(ck * exp(1i*k*w0*t) *
H(j*k*w0));
    end
    yM2(ctr) = A*exp(-1*t/tau) + yMsteady(ctr);
    ctr = ctr + 1;
end

t = 0:0.01:24*T;
figure; plot(t, fM2, 'b', t, yMsteady, 'r', t, yM2, 'k');

```

```
title('half-wave rectifier, M = 30, \tau = 10T');
```

```
%PART C
%M=30, tau=5T
num=[1];
den=[5.*T,1];
H = tf(num,den);
ym_lsim = lsim(H, fM1, t);
e5 =norm(yM1 - ym_lsim')
```

```
%M=30, tau=10T
num=[1];
den=[10.*T,1];
H = tf(num,den);
ym_lsim = lsim(H,fM2, t);
e10 = norm(yM2 - ym_lsim')
```

```
P =
```

```
@(s)w0*(1+exp(-s*T/2))/(s^2+w0^2)
```

```
C =
```

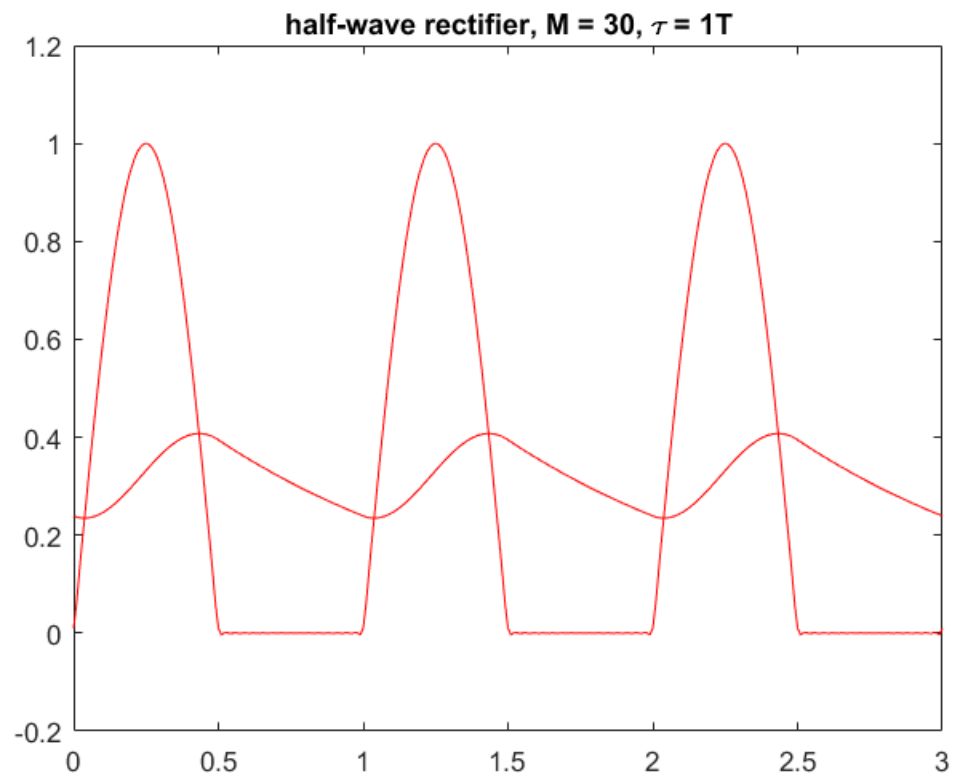
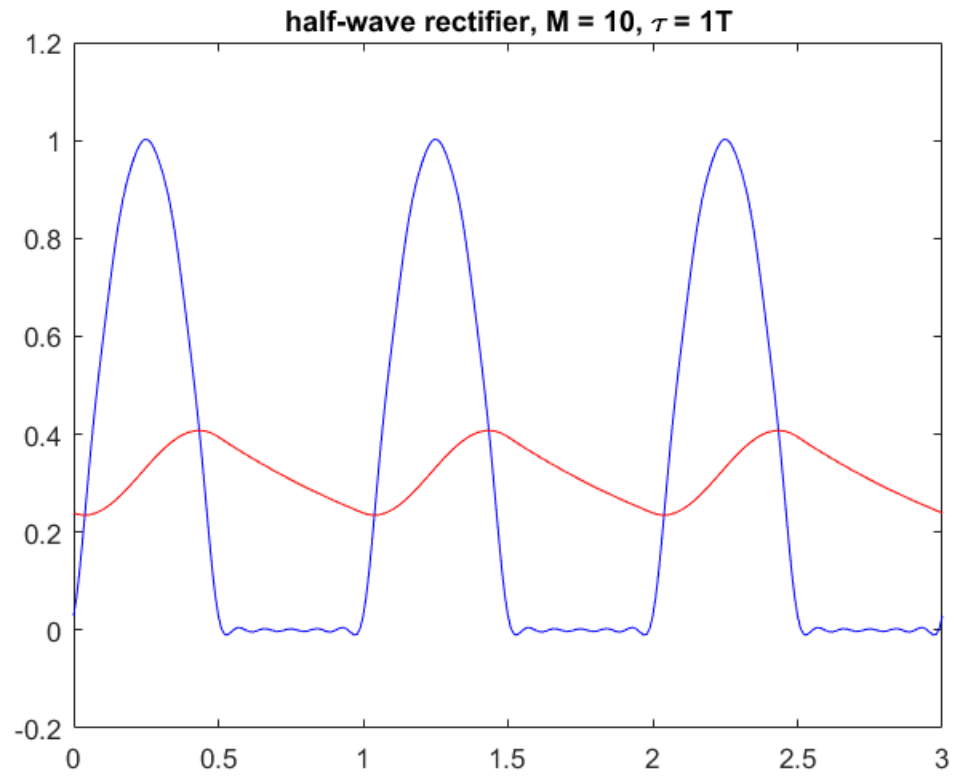
```
@(k)(1/T)*P(j*k*w0)
```

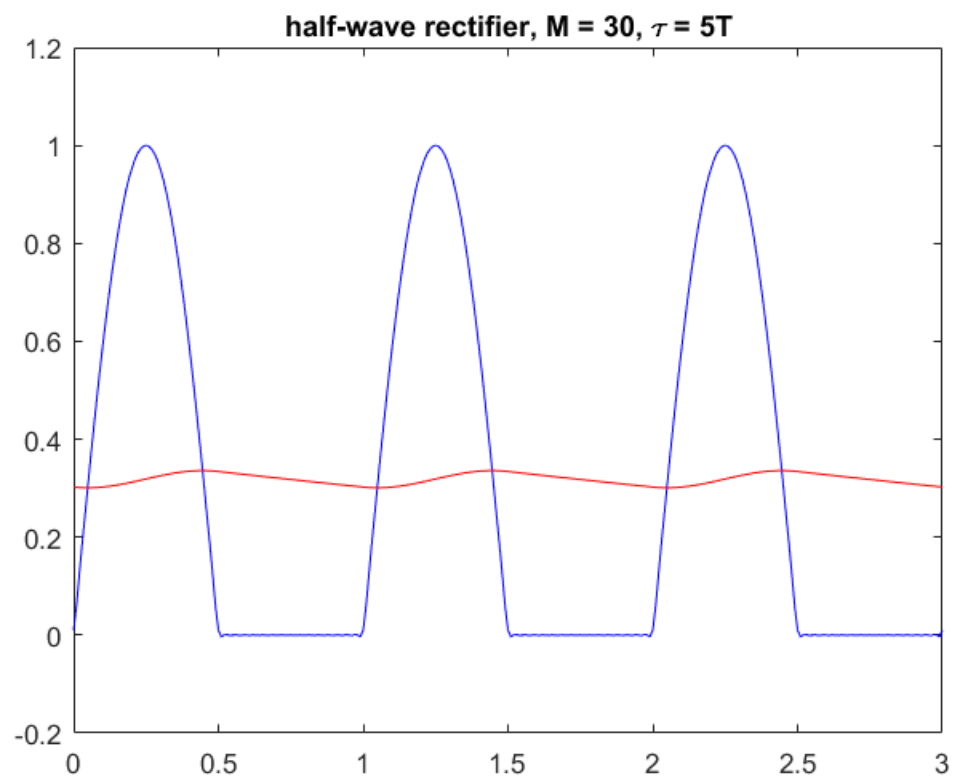
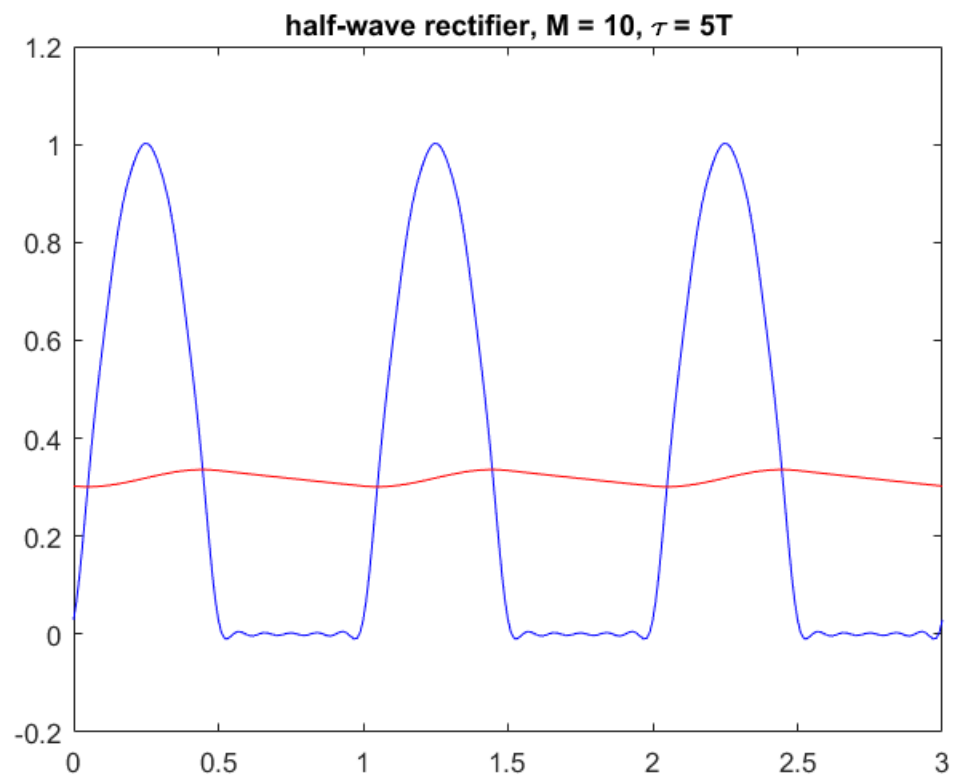
```
e5 =
```

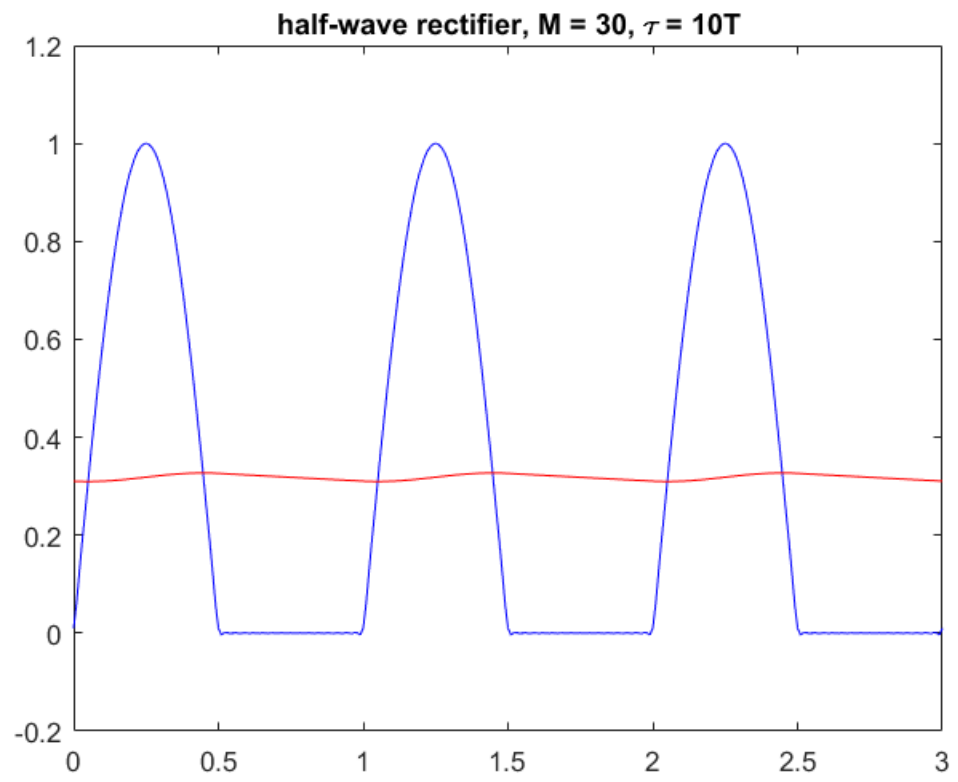
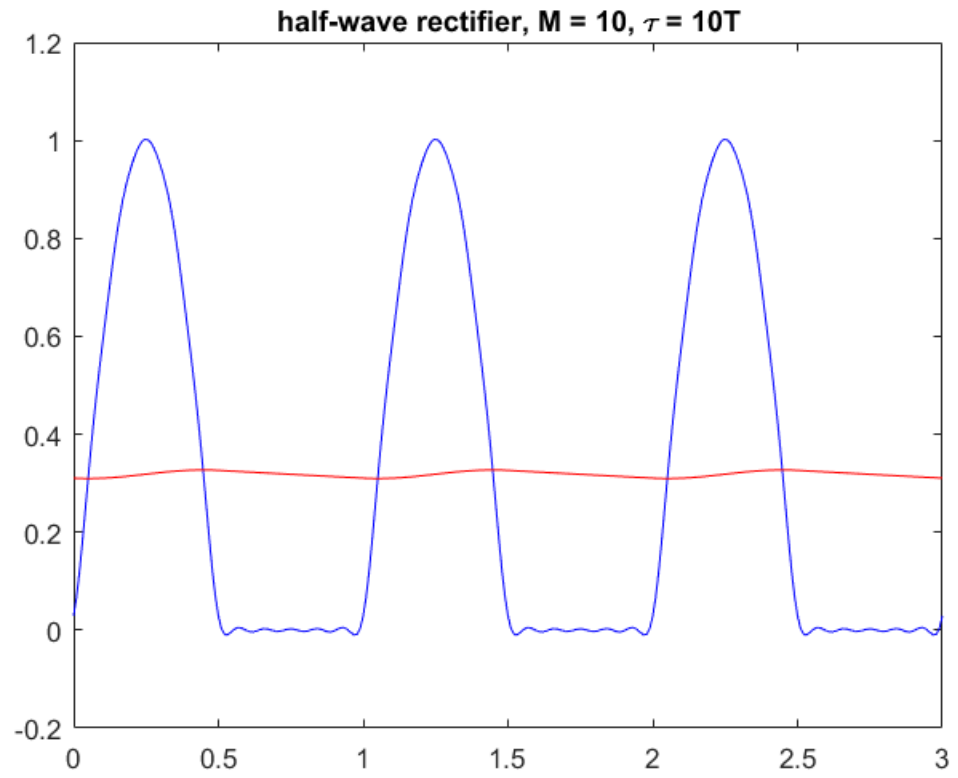
```
2.6761e-04
```

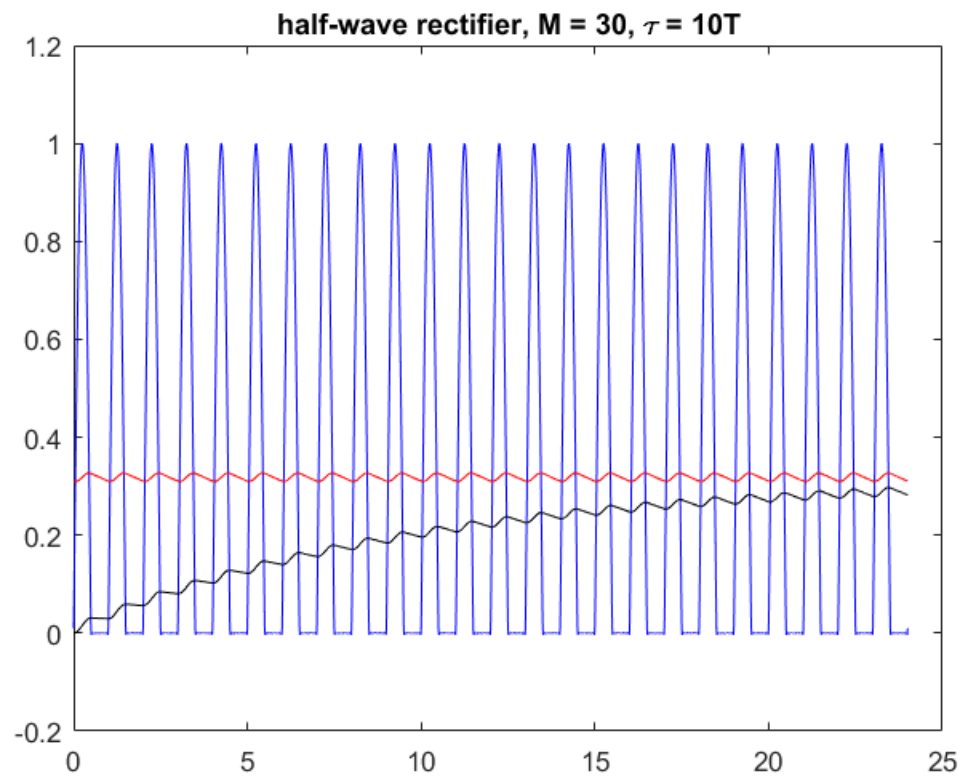
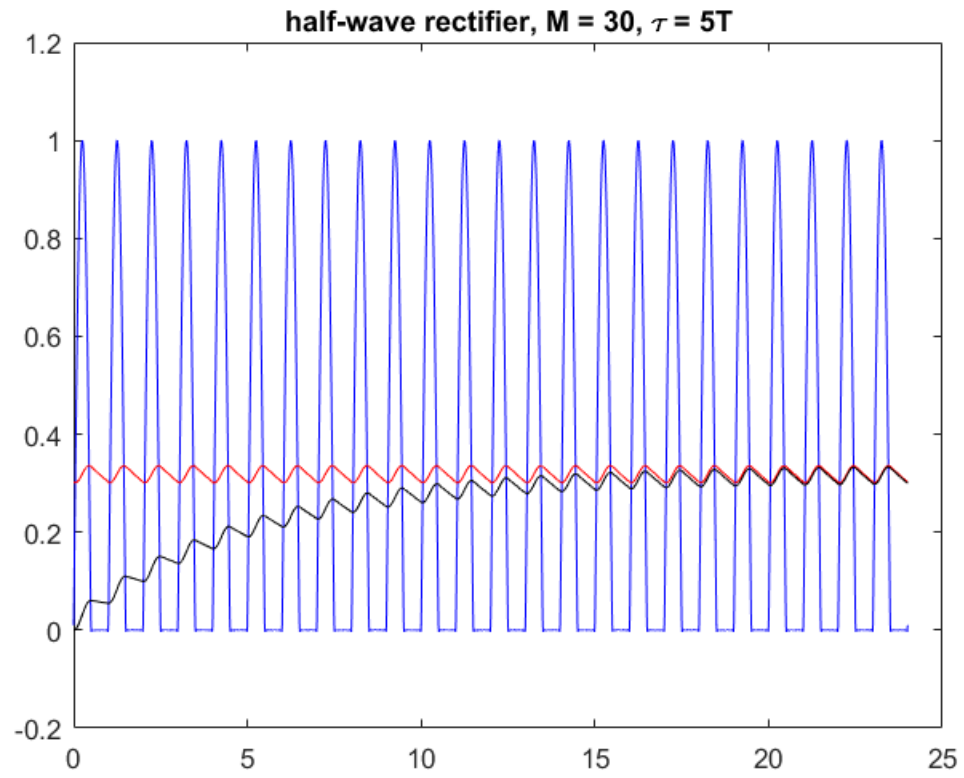
```
e10 =
```

```
1.4003e-04
```









PROBLEM 2: FIR DIGITAL FILTER DESIGN USING THE FOURIER SERIES METHOD

```
clear; clc;

%Given a signal input x(t) of length L which is sum of desired signal
s(n)
%and interference v(n), the x(t) is stripped of its interference
signal
%v(t) by passing it through a BPF, with values of omega specified
within
%the lab (code). The filter itself is designed using fourier series
%analysis in conjunction with a Hamming window.

%PART A
n = 1:199;
om0 = 0.2*pi; om1 = 0.1*pi; om2 = 0.3*pi;
omA = 0.15*pi; omB = 0.25*pi;

s = @(n) sin(om0 * n);
v = @(n) sin(om1 * n) + sin(om2 * n);
x = @(n) s(n) + v(n);

figure; plot(n,x(n),'r',n,s(n),'k:');
title('x(n) and s(n)');

%PART B
M = 75;
hn = (0.54 + 0.46*cos(pi*(n-M)/M)).*((omB)*sinc(omB*(n-M)/pi) -
(omA)*sinc(omA*(n-M)/pi))/pi;
figure; plot(n, filter(hn,1,x(n)), 'r', n, s(n), 'k:');
title('s(n) and filtered x(n), 2M = 150');

%PART C
figure; plot(n, filter(hn,1,v(n)), 'r', n, v(n), 'k:');
title('v(n) and filtered v(n), 2M = 150');

%PART D
a = [0.1, 0.1]; b = [0.2, 0.2];
c = [0.3, 0.3]; d = [0, 0.16];
e = [-100, -87.5];

w = pi * linspace(0, 0.4, 801);

hh = abs(freqz(hn, 1, w));
hhdB = mag2db(abs(hh));

figure;
plot(w/pi, hh, 'b-', a, d, 'k-', b, d, 'k-', c, d, 'k-', ...
'linewidth', 1.5, 'markersize', 20);
xlabel('\omega (\pi)'); grid on;
title('Magnitude Response |H(\omega)|, 2M = 150');
```

```

figure;
plot(w/pi, hhdB, 'b-', a, d, 'k-', b, d, 'k-', c, d, 'k-', ...
      'linewidth', 1.5, 'markersize', 20);
xlabel('\omega (\pi)'); grid on;
title('Magnitude Response |H(\omega)| in dB, 2M = 150');

clear;

%PART E
%PART A REPEATED
n = 1:199;
om0 = 0.2*pi; om1 = 0.1*pi; om2 = 0.3*pi;
omA = 0.15*pi; omB = 0.25*pi;

s = @(n) sin(om0 * n);
v = @(n) sin(om1 * n) + sin(om2 * n);
x = @(n) s(n) + v(n);

figure; plot(n,x(n),'r',n,s(n),'k:');
title('x(n) and s(n)');

%PART B REPEATED
M = 100;
hn = (0.54 + 0.46*cos(pi*(n-M)/M)).*((omB)*sinc(omB*(n-M)/pi) -
      (omA)*sinc(omA*(n-M)/pi))/pi;
figure; plot(n, filter(hn,1,x(n)), n, s(n));
title('s(n) and filtered x(n), 2M = 200');

%PART C REPEATED
figure; plot(n, filter(hn,1,v(n)),'r-', n, v(n), 'k:');
title('v(n) and filtered v(n), 2M = 200');

%PART D REPEATED
a = [0.1, 0.1]; b = [0.2, 0.2];
c = [0.3, 0.3]; d = [0, 0.16];
e = [-100, -87.5];

w = pi * linspace(0, 0.4, 801);

hh = abs(freqz(hn, 1, w));
hhdB = mag2db(abs(hh));

figure;
plot(w/pi, hh, 'b-', a, d, 'r-', b, d, 'r-', c, d, 'r-', ...
      'linewidth', 1.5, 'markersize', 20);
set(gca, 'XTick', 0:0.1:0.4, 'XLim', [0, 0.4]);
set(gca, 'YTick', 0:0.2:1, 'XLim', [0, 1]);
xlabel('\omega in units of (\pi)'); grid on;
title('Magnitude Response |H(\omega)|, 2M = 200');

figure;
plot(w/pi, hhdB, 'b-', a, d, 'r-', b, d, 'r-', c, d, 'r-', ...
      'linewidth', 1.5, 'markersize', 20);

```

```

set(gca, 'XTick', 0:0.1:0.4, 'XLim', [0, 0.4]);
%set(gca, 'YTick', -100:25:0, 'XLim', [-100, 0]);
xlabel('\omega in units of (\pi)'); grid on;
title('Magnitude Response |H(\omega)| in dB, 2M = 200');

clear all;

%PART F
a = [0.1, 0.1]; b = [0.2, 0.2];
c = [0.3, 0.3]; d = [0, 0.16];
e = [-100, -87.5];

omA = 0.15*pi; omB = 0.25*pi;
w = pi * linspace(0, 0.4, 801);
n = 1:199;

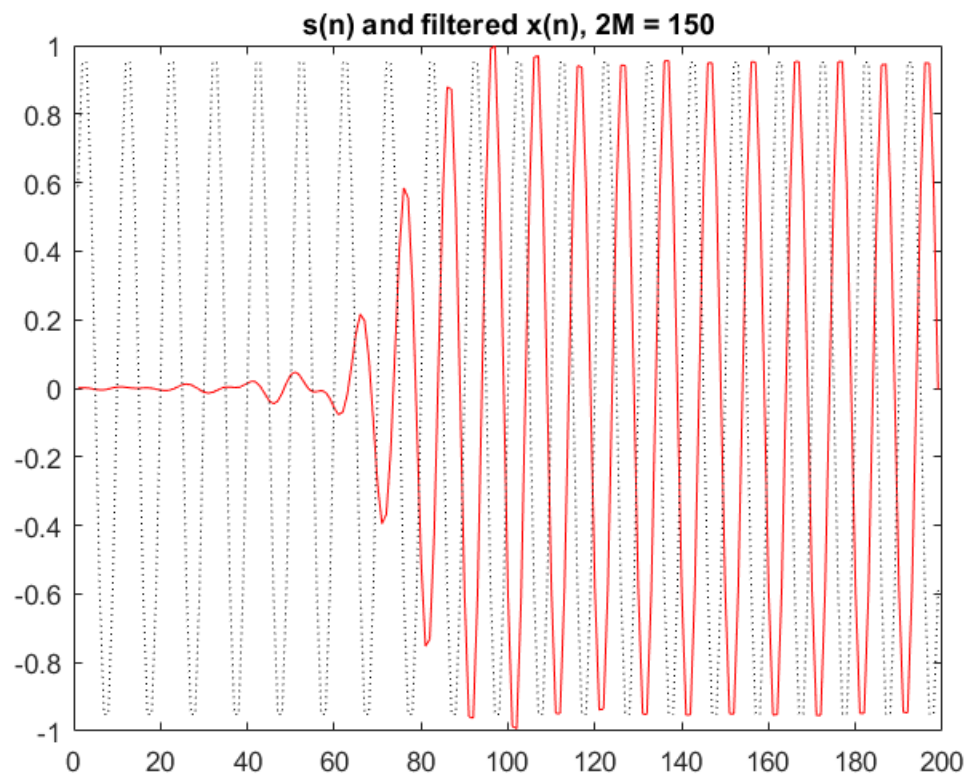
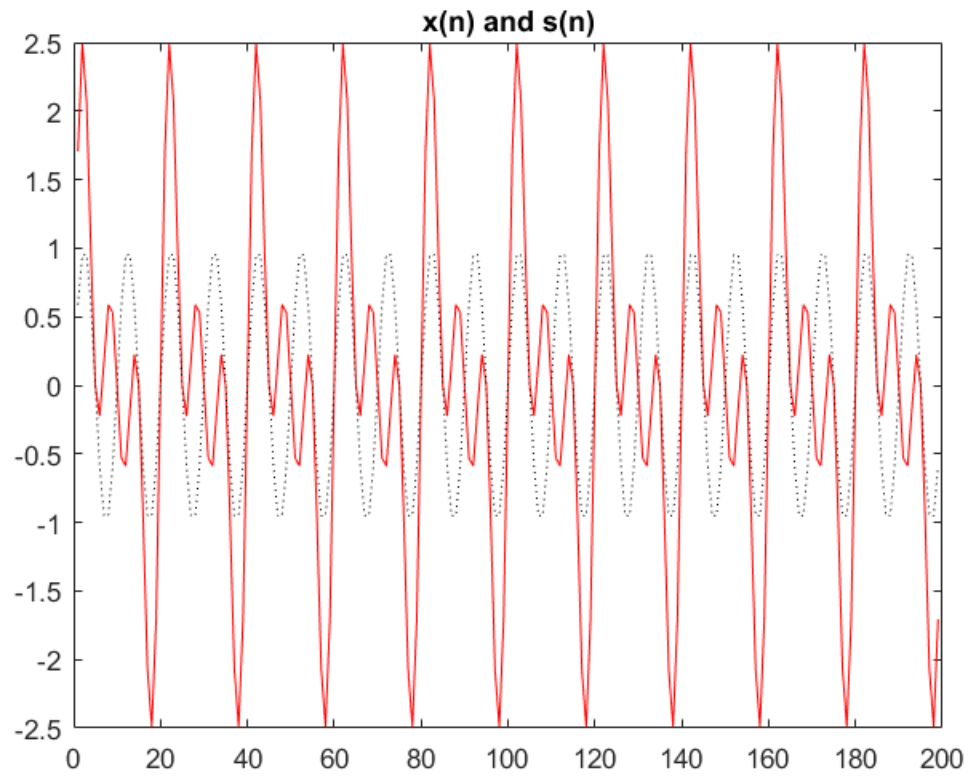
M1 = 75;
%hn1 = (cos(pi*(n-M1)/M1)).*((omB)*sinc(omB*(n-M1)/pi) -
    (omA)*sinc(omA*(n-M1)/pi))/pi;
hn1 = ((omB)*sinc(omB*(n-M1)/pi) - (omA)*sinc(omA*(n-M1)/pi))/pi;
hh1 = abs(freqz(hn1, 1, w));
hhdB1 = mag2db(abs(hh1));

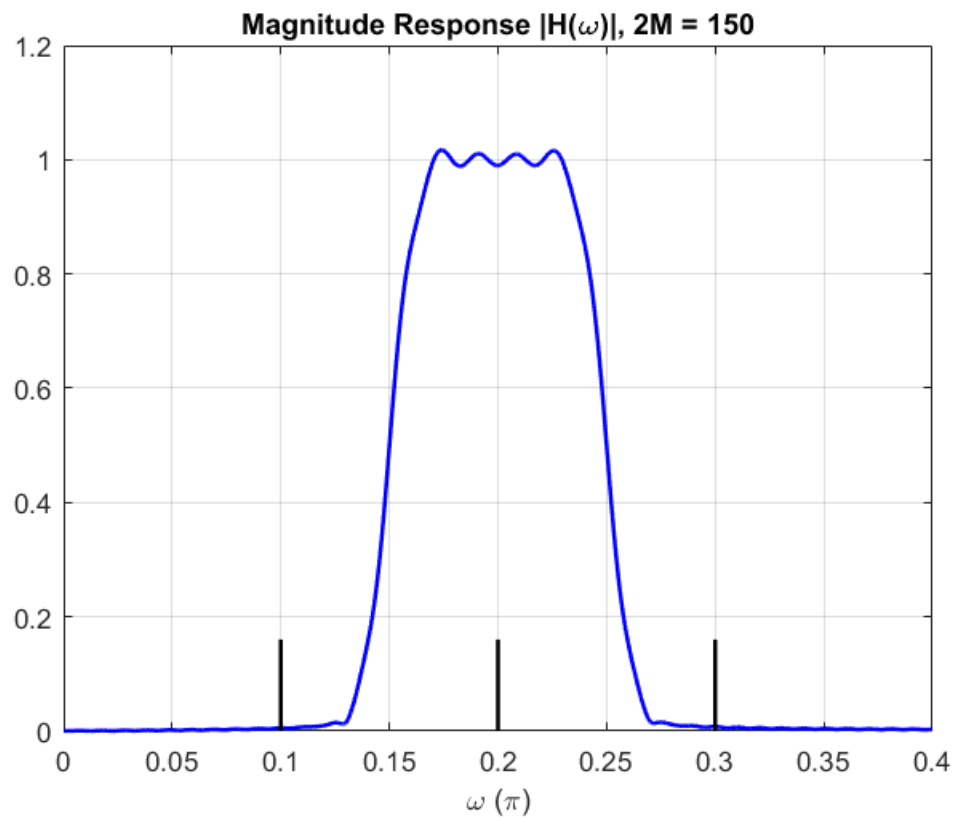
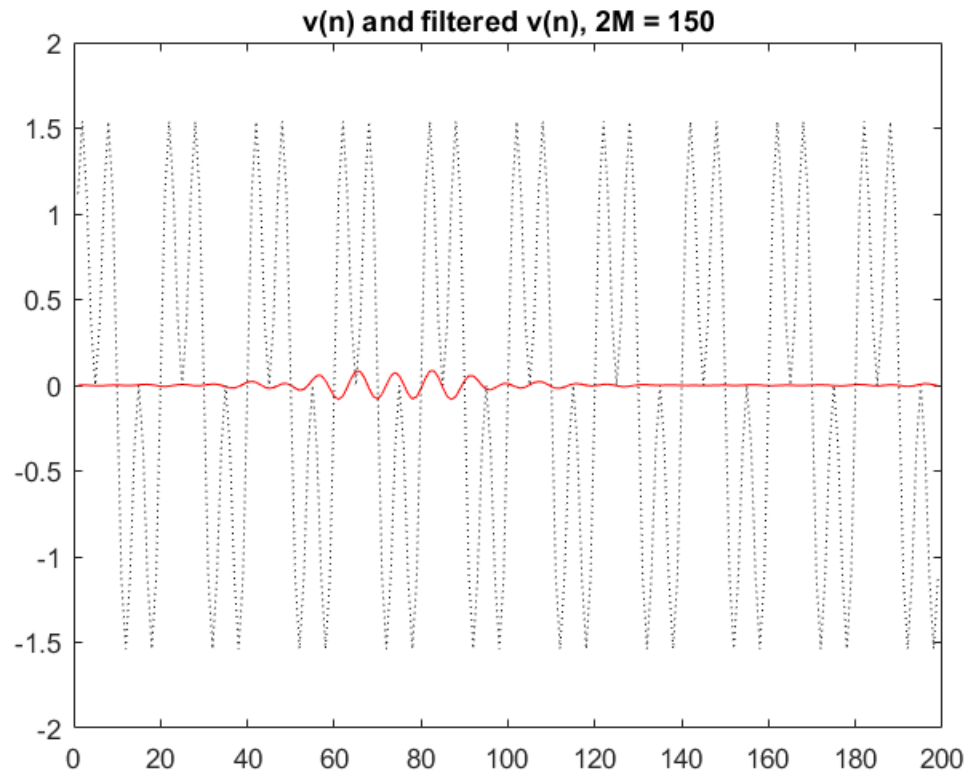
figure;
plot(w/pi, hh1, 'b-', a, d, 'r-', b, d, 'r-', c, d, 'r-', ...
    'linewidth', 1.5, 'markersize', 20);
set(gca, 'XTick', 0:0.1:0.4, 'XLim', [0, 0.4]);
set(gca, 'YTick', 0:0.2:1, 'XLim', [0, 1]);
xlabel('\omega (\pi)'); grid on;
title('Magnitude Response Gibbs |H(\omega)|, 2M = 150');

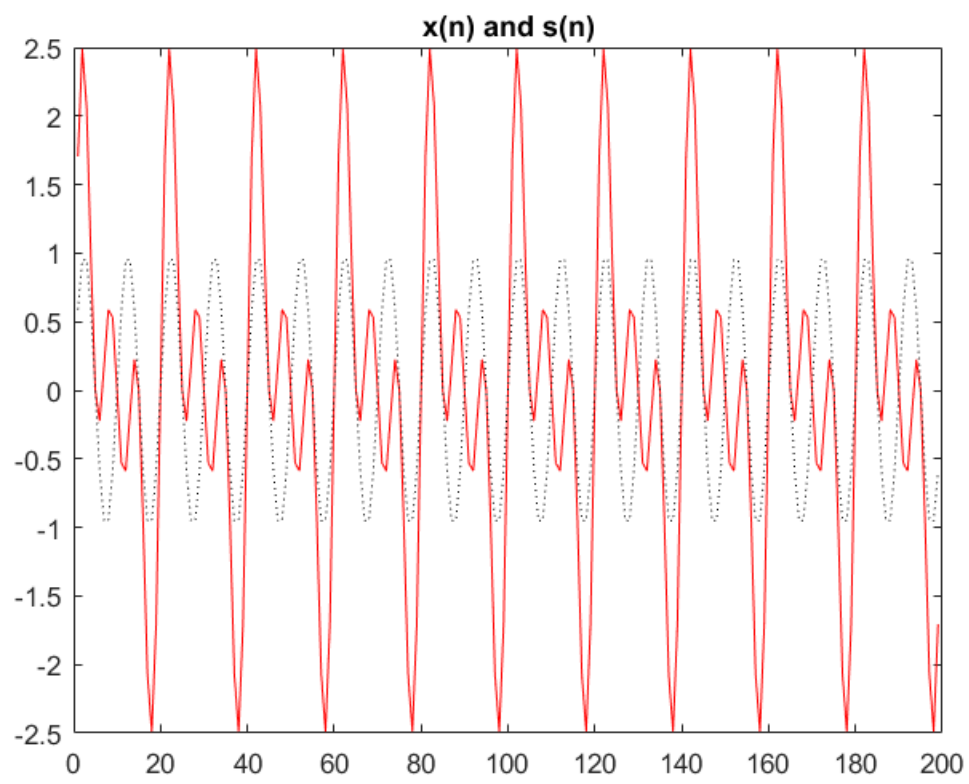
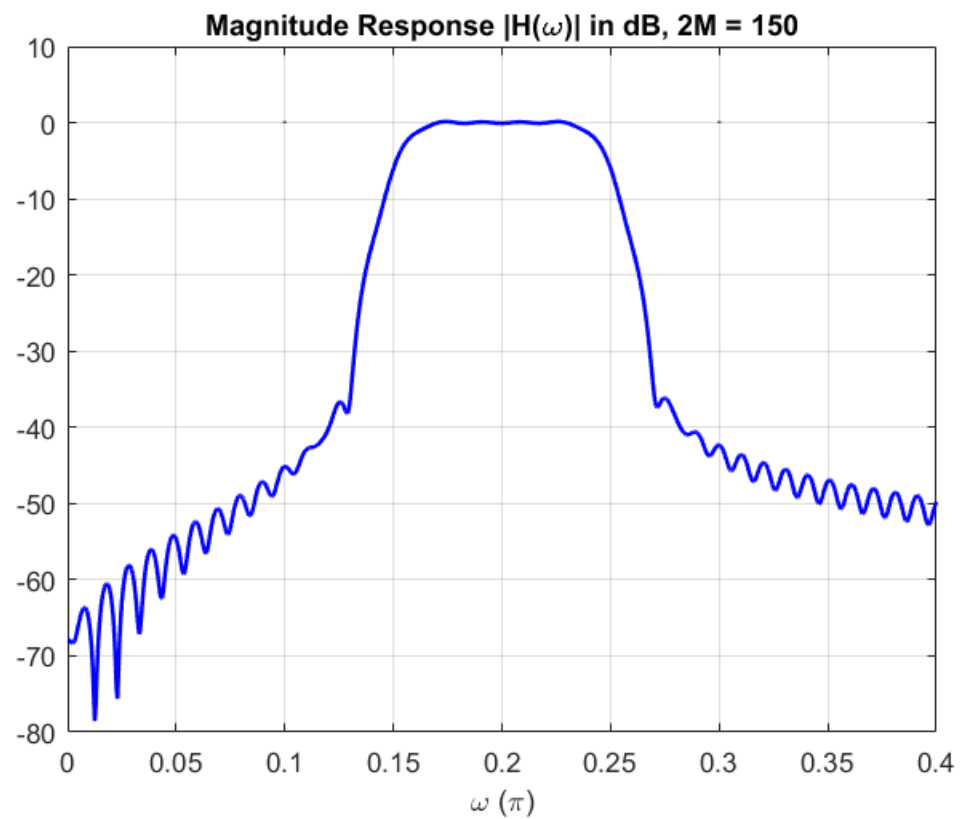
M2 = 100;
%hn2 = (cos(pi*(n-M2)/M2)).*((omB)*sinc(omB*(n-M2)/pi) -
    (omA)*sinc(omA*(n-M2)/pi))/pi;
hn2 = ((omB)*sinc(omB*(n-M2)/pi) - (omA)*sinc(omA*(n-M2)/pi))/pi;
hh2 = abs(freqz(hn2, 1, w));
hhdB1 = mag2db(abs(hh2));

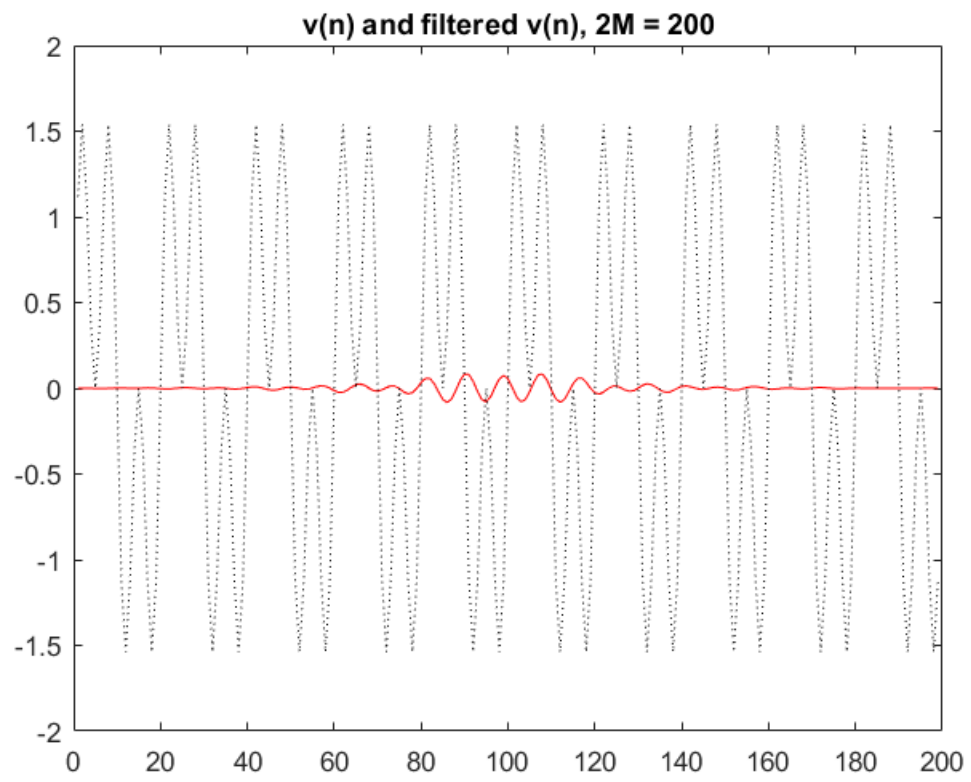
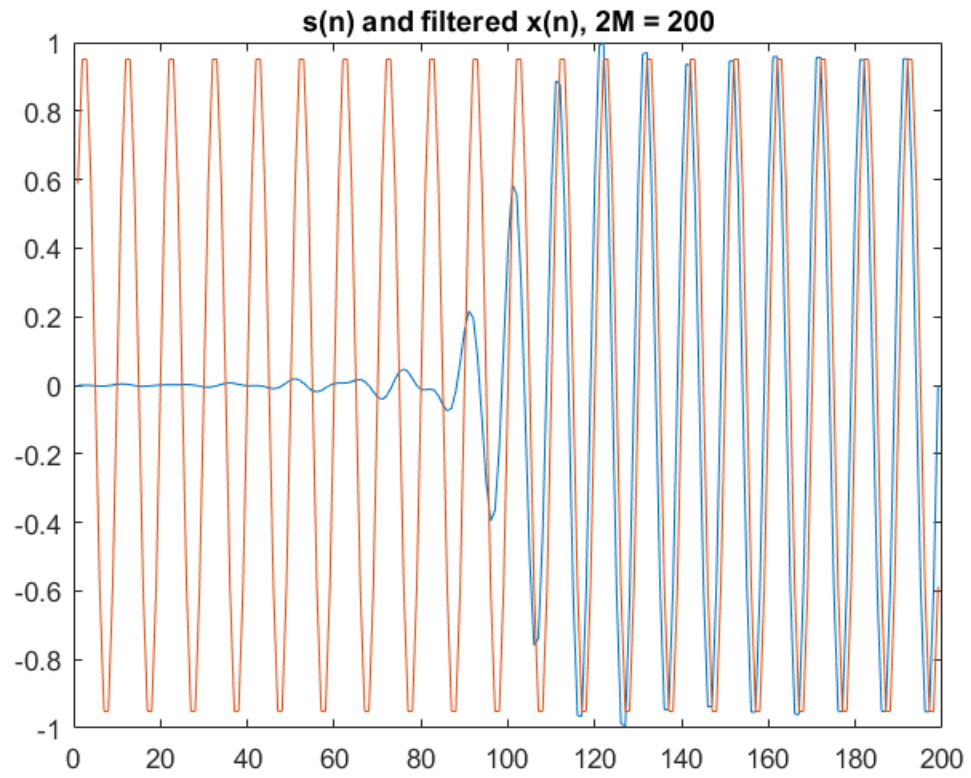
figure;
plot(w/pi, hh2, 'b-', a, d, 'r-', b, d, 'r-', c, d, 'r-', ...
    'linewidth', 1.5, 'markersize', 20);
%plot(w/pi, hh2, 'b-', 'linewidth', 1.5, 'markersize', 20);
set(gca, 'XTick', 0:0.1:0.4, 'XLim', [0, 0.4]);
set(gca, 'YTick', 0:0.2:1, 'XLim', [0, 1]);
%hold on; arrow1 = annotation('arrow', [0.1 0.1], [0 0.2]);
arrow1.Color = 'red';
xlabel('\omega (\pi)'); grid on;
title('Magnitude Response Gibbs |H(\omega)|, 2M = 200');

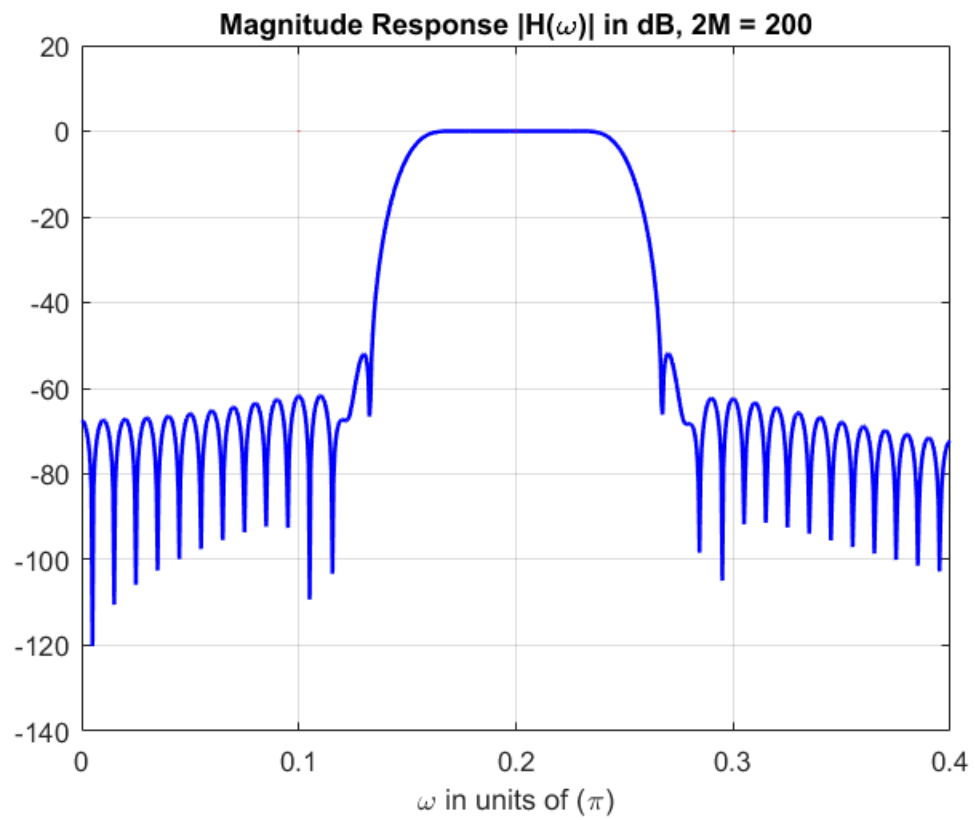
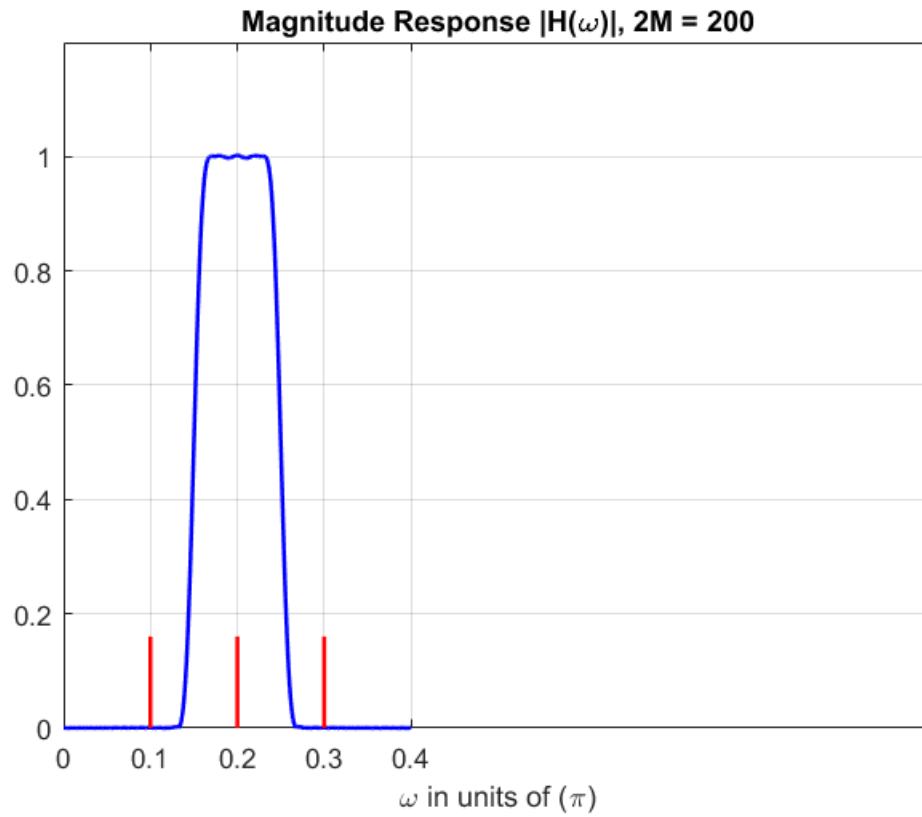
```

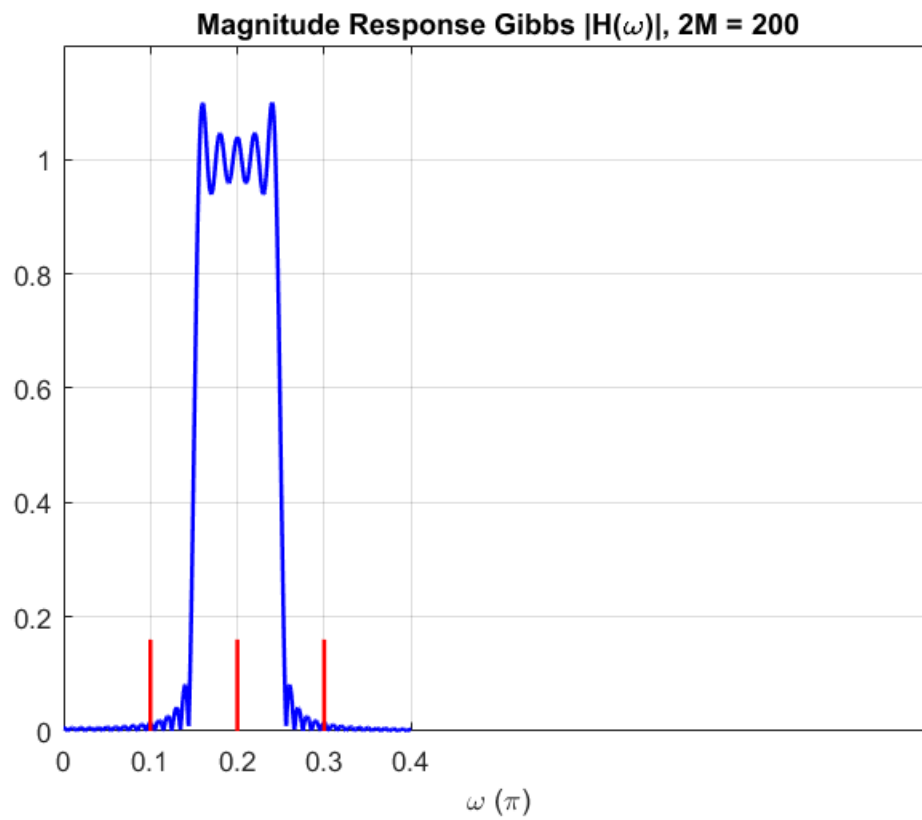
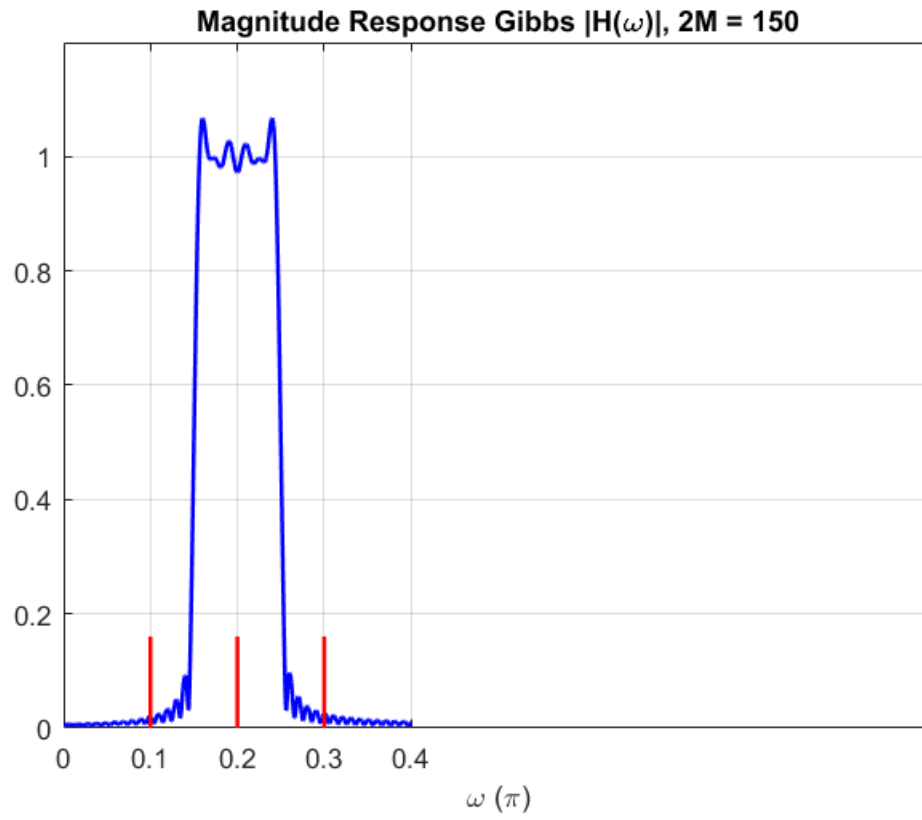












Published with MATLAB® R2016a