Lab 6: **Tasks**

Reg. No. : 2016-EE-189

Task 1

Statement:

Any arbitrary complex-valued sequence $x[n]$ can be decomposed into $x[n] = x_e[n] + x_o[n]$, where $x_e[n]$ is the conjugate-symmetric part and $x_o[n]$ is the conjugate-antisymmetric part of $x[n]$, and they're given by,

$$x_e[n] = \frac{1}{2}\left(x[n] + x^*[-n]\right) \qquad \text{and} \qquad x_o[n] = \frac{1}{2}\left(x[n] - x^*[-n]\right)$$

Question (1):

Modify the $\text{evenodd}()$ function so that it accepts an arbitrary sequence and decomposes it into its conjugate-symmetric (CA) and conjugate-antisymmetric (CAS) components.

Answer (1):

The code of the modified $\text{evenodd}()$ function, that decomposes given sequence into its CS and CAS components, is given below.

```
% <function>
% signal decomposition into conjugate-symmetric (xe[n]) and
% conjugate-antisymmetric (xo[n]) parts
%
% <syntax>
% [xe, xo, m] = evenodd(x, n)
%
% <I/O>
% x = input sequence x[n] (over DT points 'n')
% xe = conjugate-symmetric (CS) part of x[n], given by xe[n] = (x[n]+x*[-n])/2
% xo = conjugate-antisymmetric (CAS) part of x[n], given by xo[n] = (x[n]-x*[-n])/2

function [xe, xo, m] = evenodd(x, n)

    % obtain the time-reversed version of x[n] i.e. x[-n]
    [xflp, nflp] =  sigfold(x, n);

    % obtain the conjugate x*[-n]
    xflpc =  conj(xflp);
```

```
    % conjugate-symmetric part of x[n], xe[n] = (x[n]+x*[-n])/2
    [xe, m] = sigadd(x, n, xflpc, nflp);
    xe = 0.5*xe;

    % conjugate-antisymmetric part of x[n], xo[n] = (x[n]-x*[-n])/2
    [xo, m] = sigadd(x, n, -xflpc, nflp);
    xo = 0.5*xo;

end
```

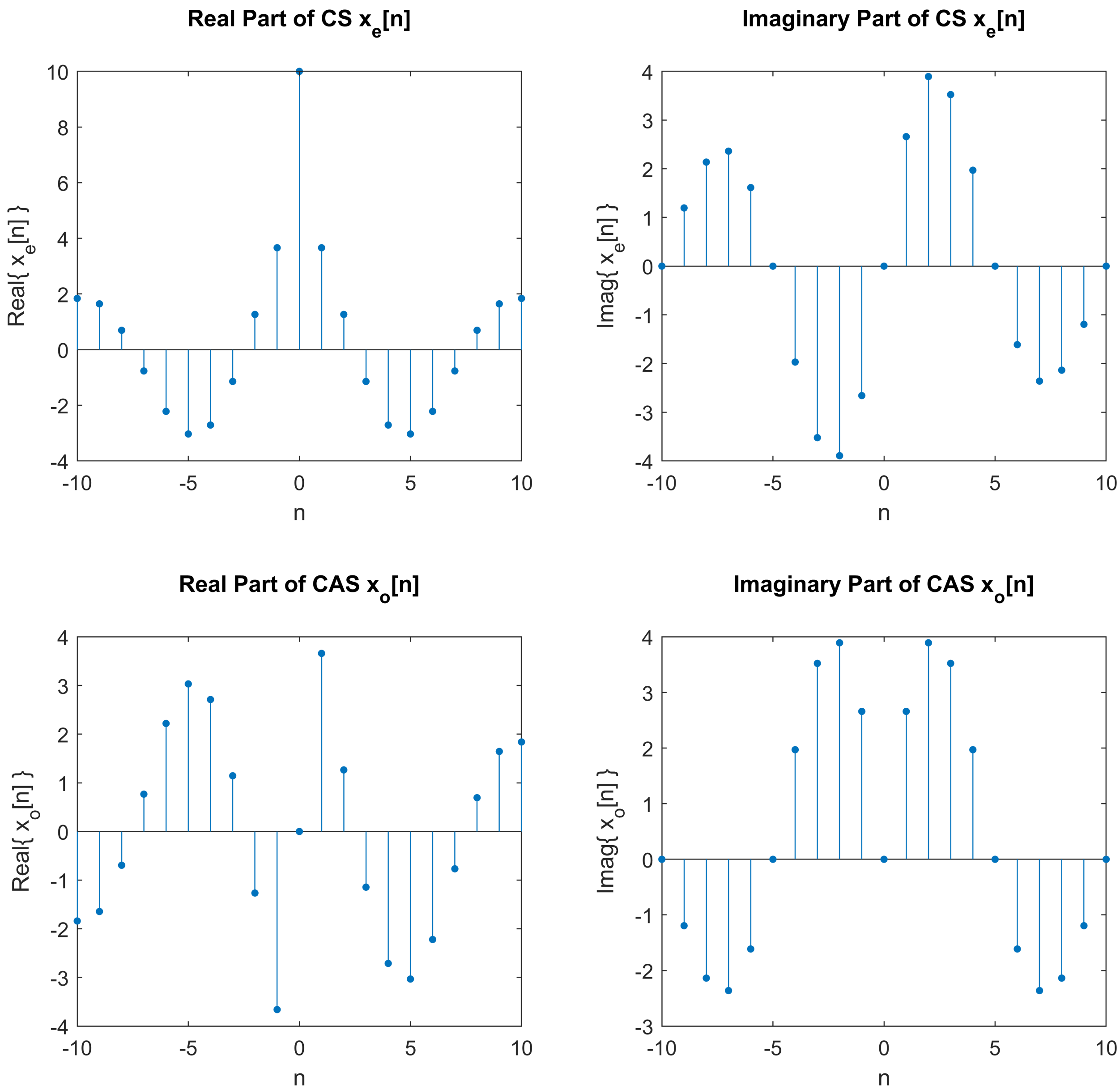
Question (2):

Decompose the sequence $x[n] = 10\exp([-0.1 + \mathrm{j}0.2\pi]n)$, $0 \le n \le 10$ into its conjugate-symmetric (CS) and conjugate-antisymmetric (CAS) components. Plot their real and imaginary parts to verify the decomposition.

Answer (2):

Following MATLAB code uses the above $\mathrm{evenodd}()$ function to decompose given $x[n]$ into its CS and CAS parts.

```
% define the sequence x[n]
n = 0:10;
x = 10*exp((-0.1 + 1i*0.2*pi)*n);

% obtain CS (xe[n]) and CAS (xo[n]) parts of x[n]
[xe, xo, m] = evenodd(x, n);

% plot real and imaginary parts of CS & CAS decomposition
fig = figure; set(fig,'Units','normalized','Position',[0 0 1 1.6]);

% plot real and imaginary parts of CS xe[n]
subplot(2, 2, 1);
stem(m, real(xe), 'filled', 'MarkerSize', 3);
xlabel('n'); ylabel('Real\{ x_e[n] \}');
title({'Real Part of CS x_e[n]';''});
subplot(2, 2, 2);
stem(m, imag(xe), 'filled', 'MarkerSize', 3);
xlabel('n'); ylabel('Imag\{ x_e[n] \}');
title({'Imaginary Part of CS x_e[n]';''});

% plot real and imaginary parts of CAS xo[n]
subplot(2, 2, 3);
stem(m, real(xo), 'filled', 'MarkerSize', 3);
xlabel('n'); ylabel('Real\{ x_o[n] \}');
title({'Real Part of CAS x_o[n]';''});
subplot(2, 2, 4);
stem(m, imag(xo), 'filled', 'MarkerSize', 3);
xlabel('n'); ylabel('Imag\{ x_o[n] \}');
title({'Imaginary Part of CAS x_o[n]';''});
```



Next task starts from the next page.

Task 2

Statement:

The operation of *signal dilation* (or *decimation* or *down-sampling*) is defined by $y[n] = x[\mathrm{nM}]$, in which the sequence $x[n]$ is down-sampled by an integer factor M.

Question (1):

Develop a MATLAB function $\mathrm{function}\ [y,\ m] = \mathrm{dnsample}(x,\ n,\ M)$ to implement the down-sampling operation.

Answer (1):

Code for $\mathrm{dnsample}()$, that downsamples a given sequence $x[n]$ by a factor M to obtain $y[m]$, is given below.

```
% <function>
% downsamples sequence x[n] by a factor M to obtain y[n] = x[nM],
% returns empty vectors if no samples exist in resultant range
%
% <syntax>
% [y, m] = dnsample(x, n, M)
%
% <I/O>
% x = input sequence x[n],  M = decimation factor,  y = output (dilated) sequence = x[nM]

function [y, m] = dnsample(x, n, M)

    % M must be a non-zero positive number (int)
    if (M <= 0); error('M must be a non-zero positive number'); end; M = round(M, 0);

    % mod(a, b): nonzero results are always negative if the divisor is negative
    diff = mod(n(1), M);

    % keeping y[0]=x[0.M]=x[0], beginning m value should be
    % and always positive if the divisor is positive
    if diff == 0
        m_b_inx = 1;                        % starts at same value as in n
    else
        m_b = n(1) + M - mod(n(1), M);
        m_b_inx = find(n == m_b);           % index in n of beginning m value
        if ~isempty(m_b_inx) m_b_inx = m_b_inx(1); end;
    end

    % decimate x[n] to get y[n]=x[nM]
    m = n(m_b_inx:M:end)/M;
    y = x(m_b_inx:M:end);

end
```

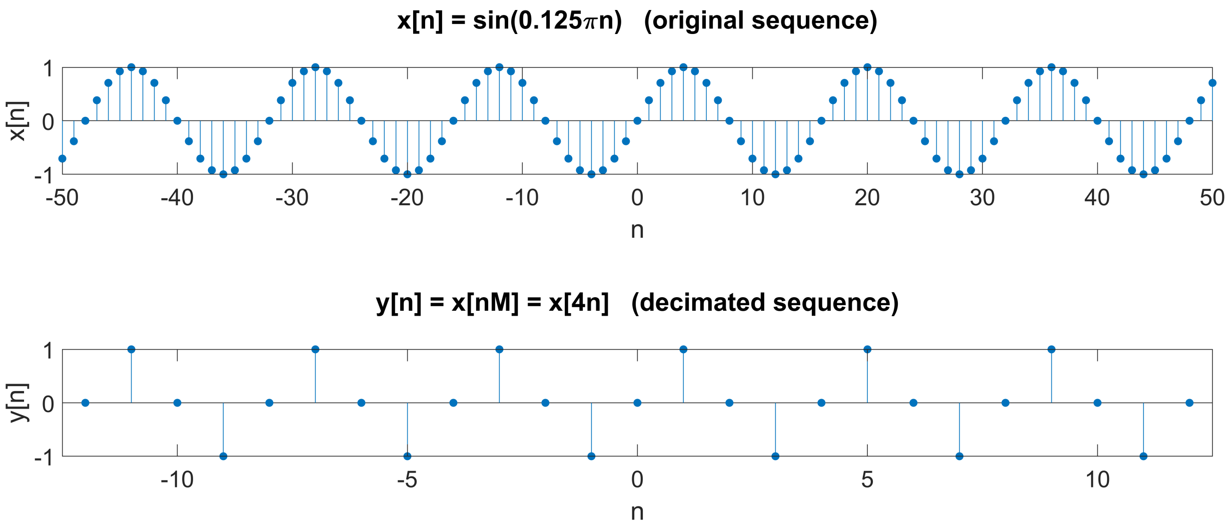
Question (2):

Generate $x[n] = \sin(0.125\pi n)$, $-50 \leq n \leq 50$. Decimate $x[n]$ by a factor of 4 to generate $y[n]$. Plot both $x[n]$ and $y[n]$, and comment on the results.

Answer (2):

Following MATLAB code uses the above $\text{dnsample}()$ function to decimate given $x[n]$ by a factor of 4.

```
% define sequence x[n]
n = -50:50;
x = sin(0.125*pi*n);

% decimate x[n] by M=4
M = 4;
[y, m] = dnsample(x, n, M);

% plot sequences x[n] and y[n]
% function called below is provided at the end of code section in Answer (3)
plotXY_t2(x, n, y, m, M, 'sin(0.125\pin)');
```



Since the decimation factor here is $M = 4$, so starting from $n = 0$, in either direction, only the fourth sample is kept, and the 3 in between are dropped. The sample value at $n = 5$ in $y[n]$ is the one at $n = 5(4) = 20$ in $x[n]$, and the value at $n = 10$ in $y[n]$ is the one at $n = 10(4) = 40$ in $x[n]$, and so on. This means that for the same sample (digital) frequency, reconstrcuted $y(t)$ could have a higher (analog) frequency than $x(t)$.

Here, since $x[n]$ has a single constant sinusoidal frequency, and the oscillation is preserved after decimation, in $y[n]$, so the frequency content (DTFT) of the signal is preserved, although scaled, after this decimation by $M = 4$. This means that the original signal (say $x(t)$) is still recoverable using the decimated $y[n]$, by using a new suitable sample-rate.

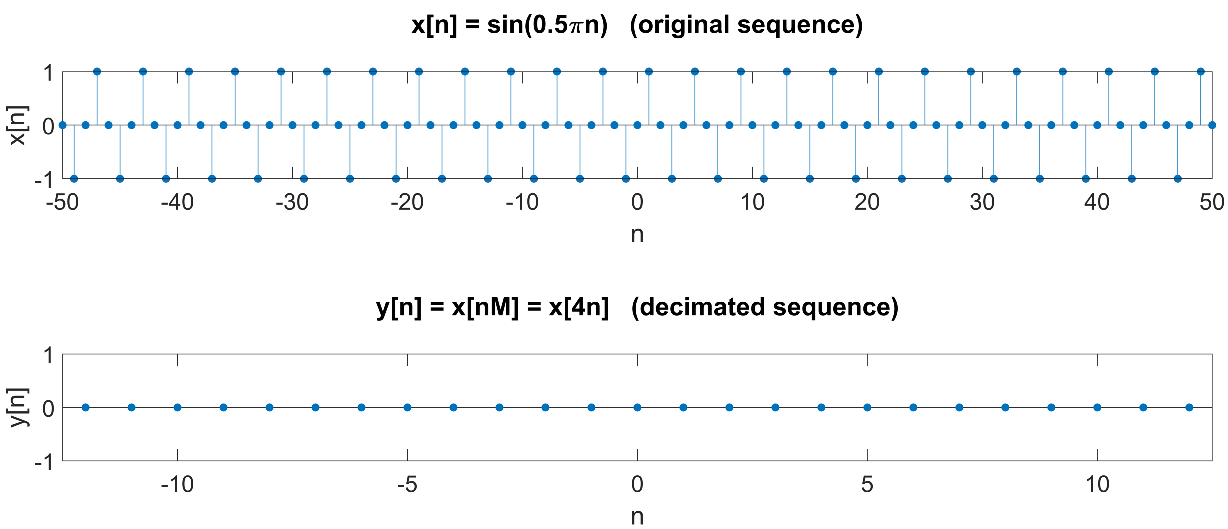
Question (3):

Repeat Q3 using $x[n] = \sin(0.5\pi n)$, $-50 \le n \le 50$. Qualitatively discuss the effect of down-sampling on signals.

Answer (3):

Following MATLAB code uses the $\mathrm{dnsample}()$ function (from Q1) to decimate given $x[n]$ by a factor of 4.

```
% define sequence x[n]
n = -50:50;
x = sin(0.5*pi*n);

% decimate x[n] by M=4
M = 4;
[y, m] = dnsample(x, n, M);

% plot sequences x[n] and y[n] (using function given below)
plotXY_t2(x, n, y, m, M, 'sin(0.5\pin)');
```




```
% <function> to plot x[n] (original) and y[n] (decimated) in 2x1 subplot
% ------------
function plotXY_t2(x, n, y, m, M, x_eq)
    fig = figure; set(fig,'Units','normalized','Position',[0 0 1.4 1]);
    subplot(2, 1, 1);
    stem(n, x, 'filled', 'MarkerSize', 5);          % plot x[n]
    xlabel('n'); ylabel('x[n]'); xlim([min(n) max(n)]);
    title({['x[n] = ', x_eq,' (original sequence)'];''}, 'FontSize', 16);
    subplot(2, 1, 2);
    stem(m, y, 'filled', 'MarkerSize', 5);          % plot y[n]
    xlabel('n'); ylabel('y[n]'); xlim([min(n)/4 max(n)/4]); ylim([min(x) max(x)]);
    title(['y[n] = x[nM] = x[', num2str(M), 'n] (decimated sequence)'], 'FontSize', 16);
end
```

Comment on the effect of down-sampling on signals:

Now (in Q3), since the oscillation is not preserved after decimation, in $y[n]$, so the original signal can no longer be recovered using the decimated $y[n]$.

For a signal $x(t)$ to be entirely recoverable from its sampled version $x[n]$, the sampling frequency $\Omega_S = 2\pi F_S$ must be twice (or more than) the maximum frequency (say Ω_N) in the time-domain signal $x(t)$, that is,

$$\Omega_S \geq 2\,\Omega_N$$

so that all the frequencies (say $-\Omega_N$ to $+\Omega_N$) in $x(t)$ are properly mapped to the DTFT range $\omega = -\pi$ to $+\pi$ in the DT version (if $\Omega_S = 2\,\Omega_N$), and any aliasing/overlapping is avoided.

Decimation by a factor of $M = 4$ essentially means that, in context of the original signal $x(t)$, the sampling-rate to obtain its (decimated) DT version has by reduced by a factor of $M = 4$. So if M is such that,

$$\frac{\Omega_S}{M} < 2\,\Omega_N$$

where Ω_S/M is the effective sample-rate for $y[n]$, then the original CT signal will not be entirely recoverable from the decimated DT signal, which we can infer is the case here (in Q3).

In cases where the Nyquist limit is still satisfied after decimation, the decimated signal is quite desirable for the speed-ups it provides in computation, compared to the greater number of unnecesarry computations in case of the undecimated sequence; as both result in the same output (CT) signal upon reconstruction.

Next task starts from the next page.

Task 3

Statement:

Consider the following finite-duration sequences, called *windows:*

$$\begin{aligned}
\text{Rectangular}: \quad & R_M[n] \;=\; 1 \text{ for } 0 \le n < M, \text{ and } 0 \text{ otherwise} \\
\text{Hanning}: \quad & C_M[n] \;=\; 0.5\left[1 - \cos\left(\frac{2\pi n}{M-1}\right)\right] R_M[n] \\
\text{Triangular}: \quad & T_M[n] \;=\; \left[1 - \frac{|M-1-2n|}{M-1}\right] R_M[n] \\
\text{Hamming}: \quad & H_M[n] \;=\; \left[0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right)\right] R_M[n]
\end{aligned}$$

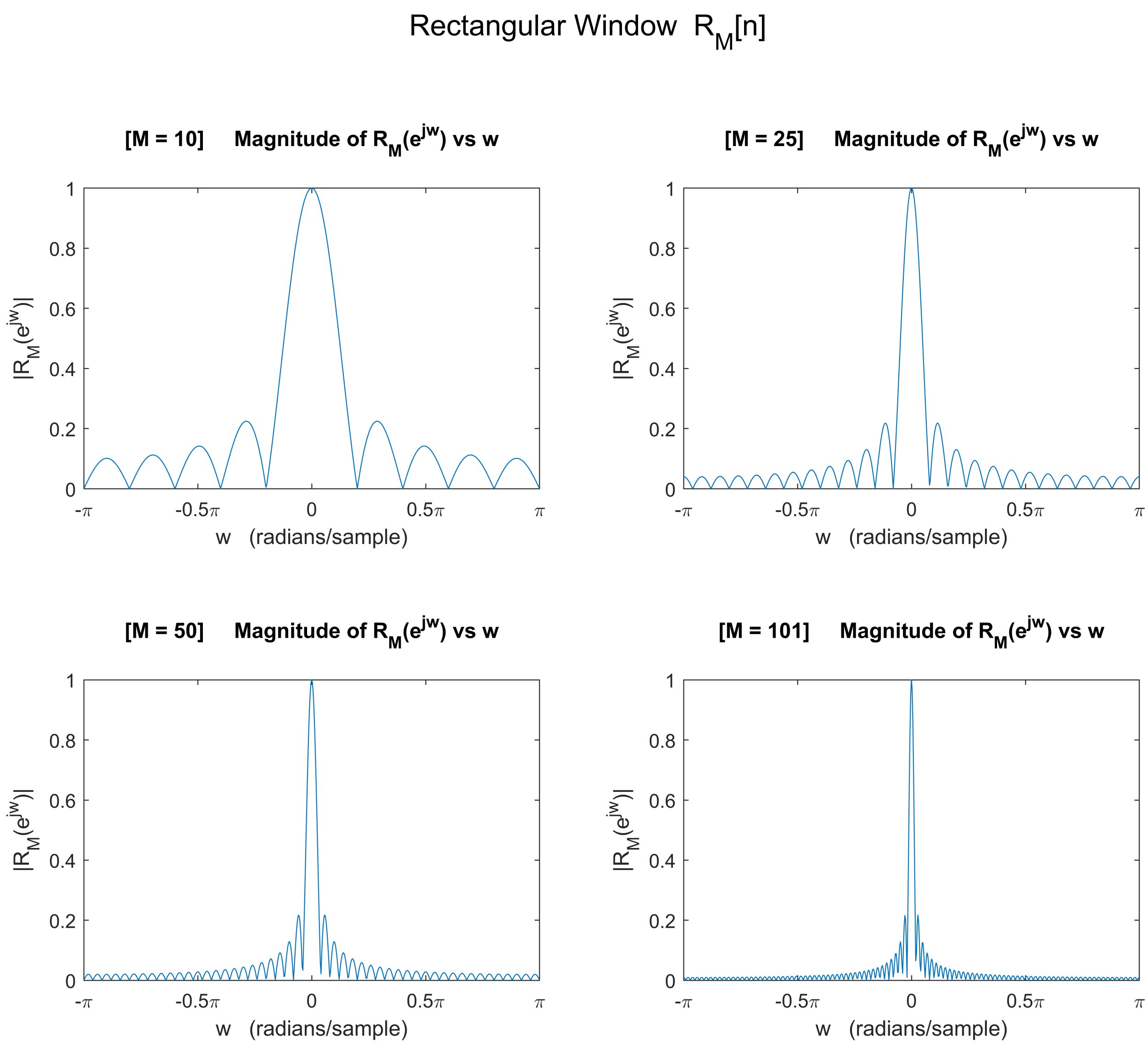
For each of these windows, determine their DTFTs for $M = 10,\ 25,\ 50,\ 101$. Scale transform values so that the maximum value is equal to 1. Plot the magnitude of the normalized DTFT over $-\pi \le \omega \le \pi$.

Study these plots and comment on their behavior as a function of M.

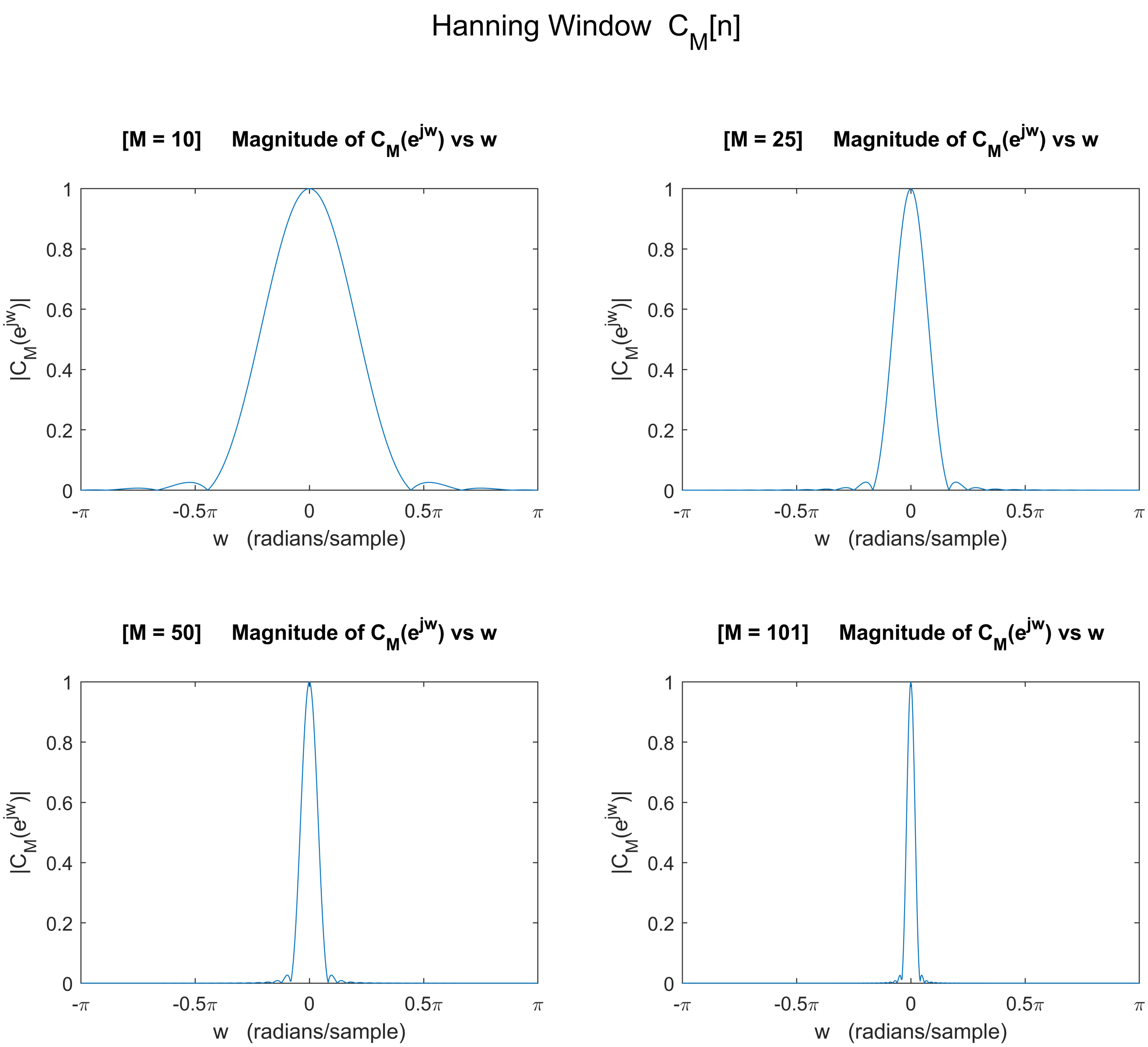
Code (with results):

The code for computing & plotting the normalized DTFT magnitudes for each M, for all the given window-sequences, is given below.

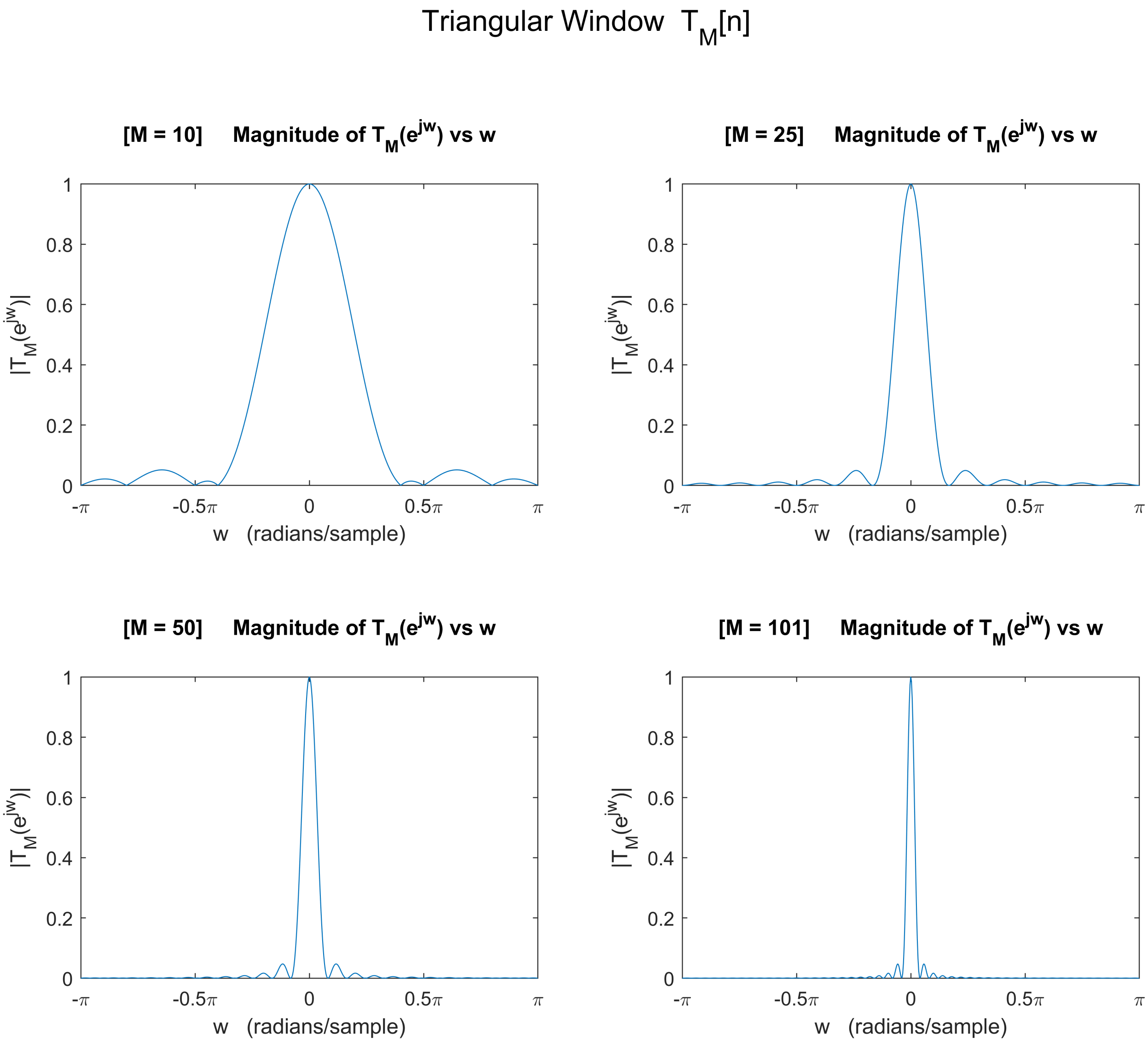
```
M = [10 25 50 101];                     % values of M for which to compute window DTFT
w = -pi:0.01:pi;                        % frequency vector for computing DTFTs of windows

% >>> window 1 <<< Rectangular: RM[n]

RM = cell(length(M), 1);                % cell-array to store RM[n] window sequences
                                        % for each M value
RMw = cell(length(M), 1);               % cell-array to store the DTFTs for each M value

for i = 1:length(M)

    % define RM[n] = 1 for 0≤n<M, 0 otherwise
    n = 0:(M(i)-1);
    RM{i} = ones(1, length(n));

    % compute the DTFT of RM[n] i.e. RM(w), using matrix-mult. method
    RMw{i} = RM{i} * exp(-1i * n' * w);

end

% plot DTFT magnitude |RM(w)| for each M (using function defined at end of this script)
plot4DTFTmag_t3(RMw, w, M, 'R', 'Rectangular Window');
```



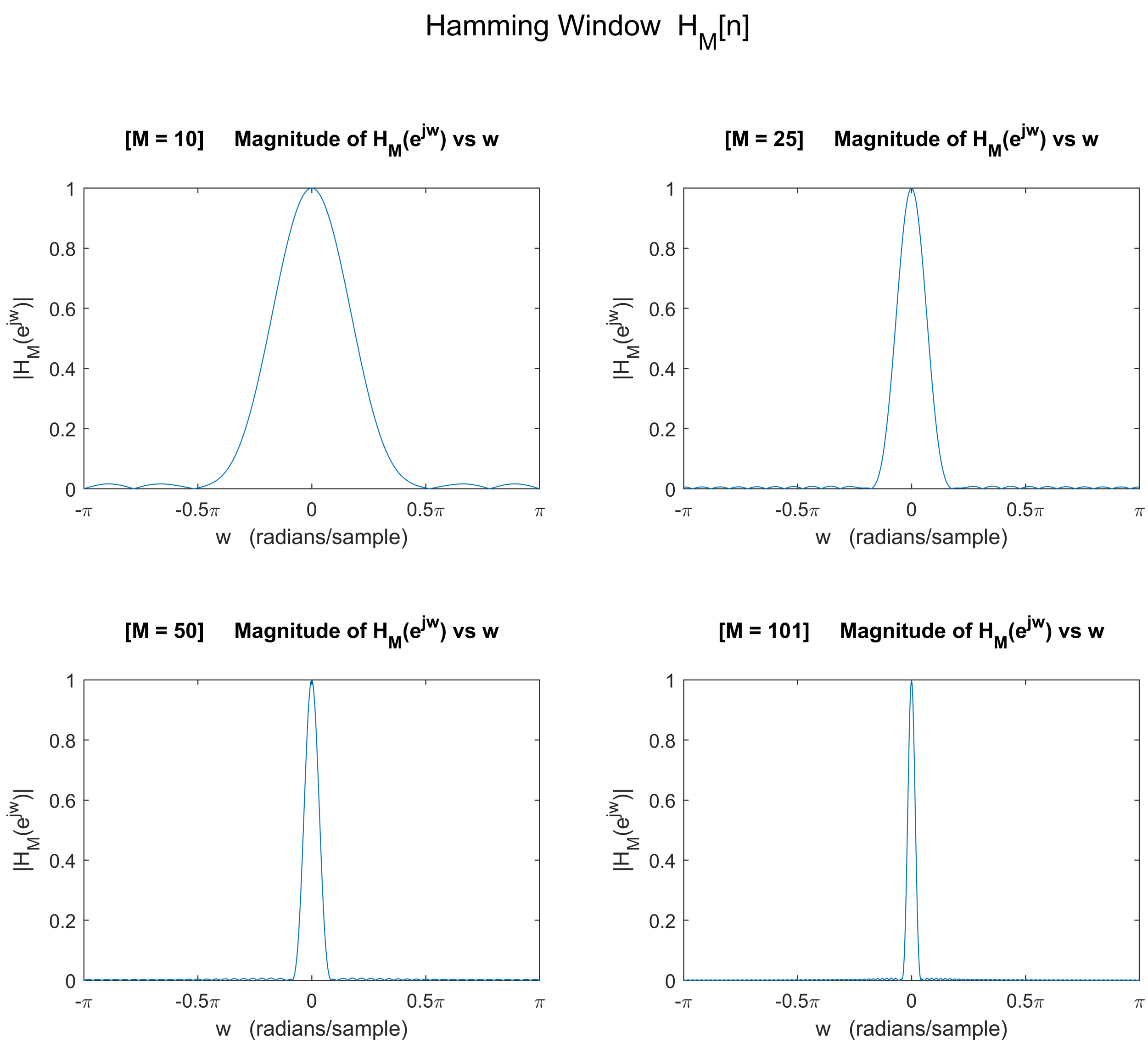

```
% >>> window 2 <<< Hanning: CM[n]

CMw = cell(length(M), 1);              % cell-array to store the DTFTs for each M value

for i = 1:length(M)

    % define CM[n] = 0.5( 1 - cos(2πn/(M-1)) ).RM[n]   (for 0≤n<M)
    n = 0:(M(i)-1);
    CM = 0.5*(1 - cos(2*pi*n/(M(i)-1))).*RM{i};

    % compute the DTFT of CM[n] i.e. CM(w), using matrix-mult. method
    CMw{i} = CM * exp(-1i * n' * w);

end

% plot DTFT magnitude |CM(w)| for each M (using function defined at end of this script)
plot4DTFTmag_t3(CMw, w, M, 'C', 'Hanning Window');
```




```
% >>> window 3 <<< Triangular: TM[n]

TMw = cell(length(M), 1);             % cell-array to store the DTFTs for each M value

for i = 1:length(M)

    % define TM[n] = (1 - |M-1-2n|/(M-1)).RM[n]   (for 0≤n<M)
    n = 0:(M(i)-1);
    TM = (1 - abs(M(i)-1-2*n)/(M(i)-1)).*RM{i};

    % compute the DTFT of TM[n] i.e. TM(w), using matrix-mult. method
    TMw{i} = TM * exp(-1i * n' * w);

end

% plot DTFT magnitude |TM(w)| for each M (using function defined at end of this script)
plot4DTFTmag_t3(TMw, w, M, 'T', 'Triangular Window');
```




```
% >>> window 4 <<< Hamming: HM[n]

HMw = cell(length(M), 1);              % cell-array to store the DTFTs for each M value

for i = 1:length(M)

    % define HM[n] = ( 0.54 - 0.46cos(2πn/(M-1)) ).RM[n]   (for 0≤n<M)
    n = 0:(M(i)-1);
    HM = (0.54 - 0.46*cos(2*pi*n/(M(i)-1))).*RM{i};

    % compute the DTFT of HM[n] i.e. HM(w), using matrix-mult. method
    HMw{i} = HM * exp(-1i * n' * w);

end

% plot DTFT magnitude |HM(w)| for each M (using function defined at end of this script)
plot4DTFTmag_t3(HMw, w, M, 'H', 'Hamming Window');
```




```
% <function> to plot DTFTs (magintude, normalized) of any window for 4 M values
% ------------
function plot4DTFTmag_t3(dtfts, w, M, wind, windn)
    fig = figure; set(fig,'Units','normalized','Position',[0 0 1 1.6]);
    sgtitle([wind_name, '  ', wind, '_M[n]']);
    for i = 1:4
        XMw_mag = abs(dtfts{i});                  % get the magnitude of XM(w)
        XMw_mag_norm = XMw_mag/max(XMw_mag);      % normalize mag. so that max value = 1

        subplot(2, 2, i);
        plot(w, XMw_mag_norm);                    % plot normalized magnitude of DTFT XM(w)
        xlabel('w   (radians/sample)'); ylabel(['|', wind, '_M' ,'(e^{jw})|']);
        ttl = {['[M = ', num2str(M(i)), '] Magnitude of ', wind, '_M(e^{jw}) vs w'];''};
        if i > 2;  ttl = [{''}; ttl]; end; title(ttl);
        setDTFTradialAxis(1);
    end
end
```

Comment on behaviour of window DTFTs for different M values:

The windows all show low-pass behaviour, with some much-smaller lobes on each side of the central lobe. For each of the windows, as M is increased, the width of the central lobe decreases, so if these windows are used as filters, then by increasing M, the cutoff frequency can be decreased. Furthermore, the cutoff is also made sharper by increasing M. Hence, by using a large M, and adding a shift to the window spectrum, a very narrow-pass filter can be made to allow only a very-small range of frequencies through the system.

The quantity and magnitude of the smaller side-lobes varies across the different windows. The rectangular window $R_M[n]$ has most prominent and most number of these side-lobes, the triangular window $T_M[n]$ has lesser, while the hanning window $C_M[n]$ has even less prominent side-lobes. The hamming window $H_M[n]$ has nearly insignificant side-lobes. The windows approach the behaviour of function $\delta(f)$ in frequency-domain for a very large M value, which can be used to single out particular frequencies, by adding some frequency-shift.

Task 4

Statement:

Show that the real part of $X(e^{j\omega})$ of a sinusoidal pulse $x[n] = \cos(\omega_0 n)\, R_M[n]$, where $R_M[n]$ is the rectangular pulse ($R_M[n] \;=\; 1 \text{ for } 0 \le n < M, \text{ and } 0 \text{ otherwise}$), is given by,

$$\begin{aligned} X_R(e^{j\omega}) \;&=\; \frac{1}{2}\cos\left\{\frac{(\omega-\omega_0)(M-1)}{2}\right\}\frac{\sin\{(\omega-\omega_0)M/2\}}{\sin\{(\omega-\omega_0)/2\}} \\ &+\; \frac{1}{2}\cos\left\{\frac{(\omega+\omega_0)(M-1)}{2}\right\}\frac{\sin\{[\omega-(2\pi-\omega_0)]M/2\}}{\sin\{[\omega-(2\pi-\omega_0)]/2\}} \end{aligned}$$

Compute and plot $X_R(e^{j\omega})$ over $[-\pi,\, \pi]$ for $\omega_0 = \pi/2$ and $M = 5,\ 15,\ 25,\ 100$. Comment on your results.

Proof for the equation of $X_R(e^{j\omega})$ (paperwork):

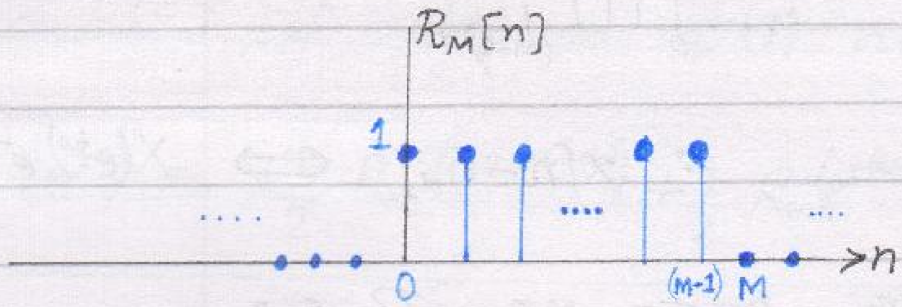
The work done on paper, deriving/showing that the equation for $X_R(e^{j\omega})$ (for given $x[n] = \cos(\omega_0 n)\, R_M[n]$) is the one shown in the Task Statement above, starts from the next page.

The work done on paper starts from the next page.

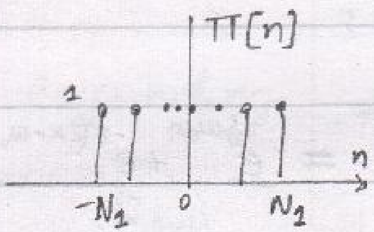
Task 4

$$x[n] = \cos(\omega_0 n) R_M[n]$$

where, $R_M[n] = \begin{cases} 1. & ; 0 \leq n < M \\ 0 & ; \text{else} \end{cases}$

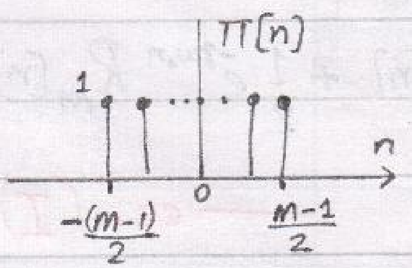


for its DTFT, first consider the gate function $\Pi[n]$ & its DTFT, as known by common DTFT pairs,



$$\Longleftrightarrow \frac{\sin\left[\omega\left(N_1 + \frac{1}{2}\right)\right]}{\sin(\omega/2)}$$

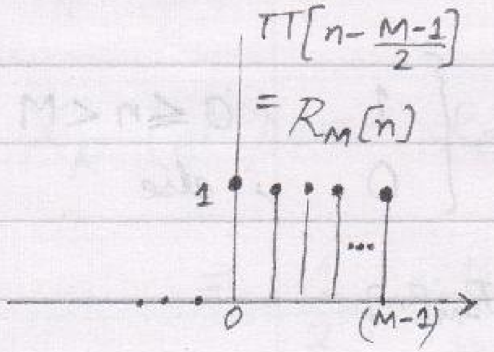
put $N_1 = \dfrac{M-1}{2}$,



$$\Longleftrightarrow \frac{\sin\left[\omega\left(\frac{M-1}{2} + \frac{1}{2}\right)\right]}{\sin(\omega/2)}$$

$$= \frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

delay by $\frac{M-1}{2}$,



$$\Pi\left[n-\frac{M-1}{2}\right] = R_M[n] \Longleftrightarrow \frac{\sin(\omega M/2)}{\sin(\omega/2)}\, e^{-j\omega\left(\frac{M-1}{2}\right)}$$

$$\because\ x[n-n_d] \Longleftrightarrow X(e^{j\omega})\, e^{-j\omega n_d}$$

so, DTFT of $R_M[n]$ is,

$$R_M(e^{j\omega}) = \frac{\sin(\omega M/2)}{\sin(\omega/2)}\, e^{-j\omega\left(\frac{M-1}{2}\right)}$$

now, also note that,

$$\cos(\omega_0 n) = \frac{e^{j\omega_0 n} + e^{-j\omega_0 n}}{2} = \frac{e^{j\omega_0 n} + e^{j(2\pi-\omega_0)n}}{2}$$

so, we have,

$$x[n] = \frac{1}{2} e^{j\omega_0 n} R_M[n] + \frac{1}{2} e^{-j\omega_0 n} R_M[n] \quad \text{—— eq. (I)}$$

Using the frequency-shifting property of DTFT, which says that,

$$e^{j\omega_0 n} x[n] \Longleftrightarrow X\left(e^{j(\omega-\omega_0)}\right)$$

the DTFT of $x[n]$ in eq. (I) is,

$$X(e^{j\omega}) = \frac{1}{2} R_M\left(e^{j(\omega-\omega_0)}\right) + \frac{1}{2} R_M\left(e^{j(\omega+\omega_0)}\right)$$

$$= \frac{1}{2} \frac{\sin\left[(\omega-\omega_0)\frac{M}{2}\right]}{\sin\left(\frac{\omega-\omega_0}{2}\right)} e^{-j(\omega-\omega_0)\left(\frac{M-1}{2}\right)}$$

$$+ \frac{1}{2} \frac{\sin\left[(\omega-(2\pi-\omega_0))\frac{M}{2}\right]}{\sin\left(\frac{\omega-(2\pi-\omega_0)}{2}\right)} e^{-j(\omega+\omega_0)\left(\frac{M-1}{2}\right)}$$

— eq. (II)

expanding further,

$$X(e^{j\omega}) = \frac{1}{2} \frac{\sin\left((\omega-\omega_0)\frac{M}{2}\right)}{\sin\left(\frac{\omega-\omega_0}{2}\right)} \left[\cos\frac{(\omega-\omega_0)(M-1)}{2} - j\sin\frac{(\omega-\omega_0)(M-1)}{2}\right]$$

$$+ \frac{1}{2} \frac{\sin\left[(\omega-(2\pi-\omega_0))\frac{M}{2}\right]}{\sin\frac{\omega-(2\pi-\omega_0)}{2}} \left[\cos\frac{(\omega+\omega_0)(M-1)}{2} - j\sin\frac{(\omega+\omega_0)(M-1)}{2}\right]$$

— eq. (III)

Taking the real part of $X(e^{j\omega})$ in eq. (III), we have,

$$X_R(e^{j\omega}) = \text{Real}\{X(e^{j\omega})\}$$

$$= \frac{1}{2}\frac{\sin\left[(\omega-\omega_0)\frac{M}{2}\right]}{\sin\left(\frac{\omega-\omega_0}{2}\right)}\left[\cos\frac{(\omega-\omega_0)(M-1)}{2}\right]$$

$$+\frac{1}{2}\frac{\sin\left[(\omega-(2\pi-\omega_0))\frac{M}{2}\right]}{\sin\left(\frac{\omega-(2\pi-\omega_0)}{2}\right)}\left[\cos\frac{(\omega+\omega_0)(M-1)}{2}\right]$$

$$= \frac{1}{2}\cos\frac{(\omega-\omega_0)(M-1)}{2}\,\frac{\sin\left[(\omega-\omega_0)M/2\right]}{\sin\left[(\omega-\omega_0)/2\right]}$$

$$+\frac{1}{2}\cos\frac{(\omega+\omega_0)(M-1)}{2}\,\frac{\sin\left[(\omega-(2\pi-\omega_0))M/2\right]}{\sin\left[(\omega-(2\pi-\omega_0))/2\right]}$$

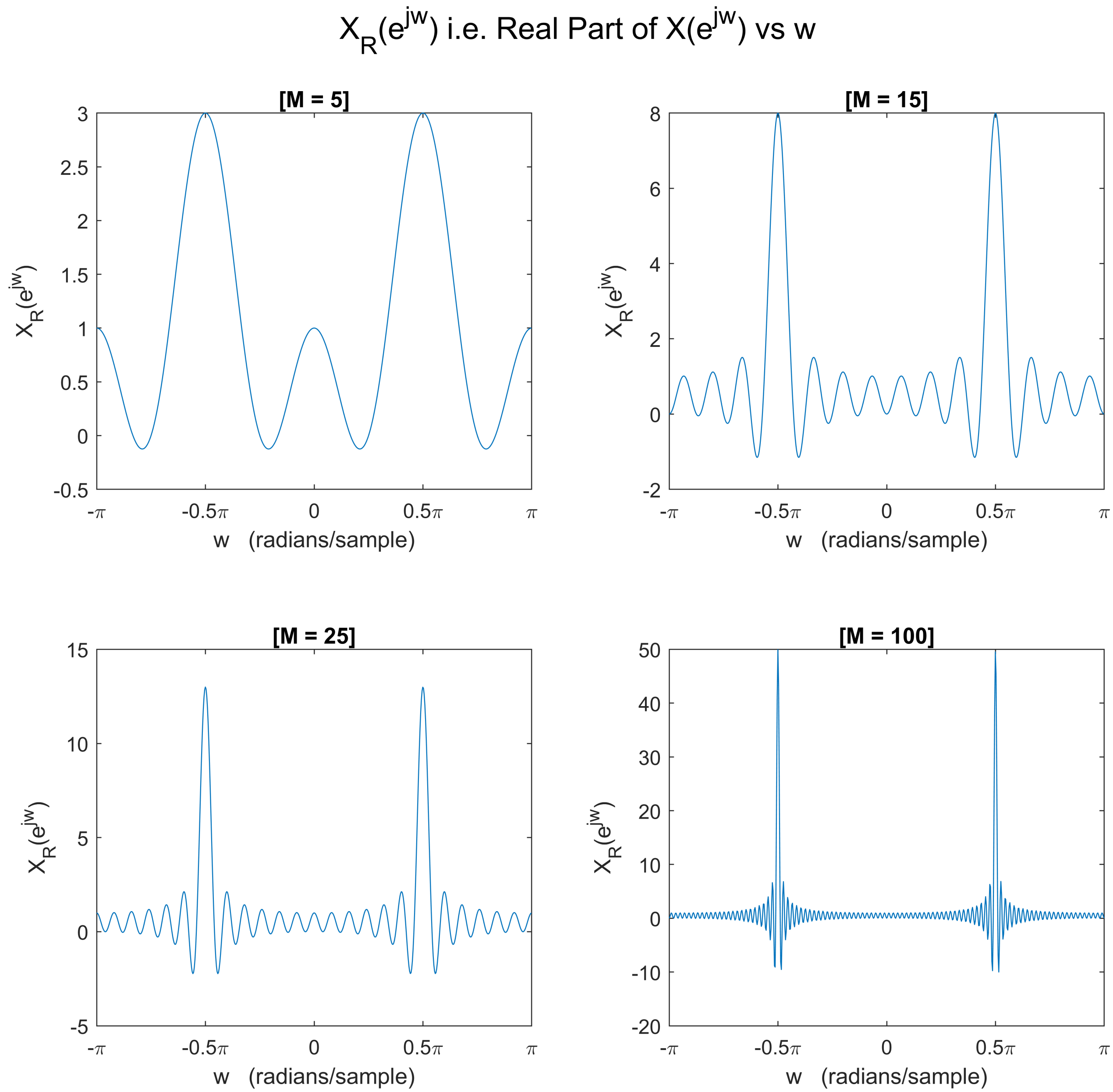
— eq. (IV)

which matches the $X_R(e^{j\omega})$ expression given in the task statement.

Plotting $X_R(e^{j\omega})$ using MATLAB:

Following code plots $X_R(e^{j\omega})$ for the various provided M values:

```
M = [5 15 25 100];                      % values of M for which to compute DTFT
w0 = pi/2;                              % given w0 value

w = -pi:0.01:pi;                        % frequency vector for computing DTFTs
XRw = cell(length(M), 1);               % cell-array to store DTFTs (real part) for each M

for i = 1:length(M)

    % define the rectangular pulse RM[n]
    n = 0:(M(i)-1);
    RM = ones(1, length(n));

    % define given x[n]
    x = cos(w0*n).*RM;

    % compute the DTFT of x[n] i.e. X(w), using matrix-mult. method
    Xw = x * exp(-1i * n' * w);

    % store the real part of DTFT i.e. XR(w)
    XRw{i} = real(Xw);

    % compare computed XR(w) with equation of XR(w) (given & derived in paperwork)
    XRw_eq = (1/2)*cos((w-w0)*(M(i)-1)/2).*sin((w-w0)*M(i)/2)./sin((w-w0)/2)...
             + (1/2)*cos((w+w0)*(M(i)-1)/2).*sin((w+w0)*M(i)/2)./sin((w+w0)/2);
    % message only displayed if results don't match
    if ~isequal(round(XRw{i}, 10), round(XRw_eq, 10))
        disp('Result Mismatch');
    end

end

% plot DTFT real part XR(w) for each M value
fig = figure; set(fig,'Units','normalized','Position',[0 0 1 1.6]);
sgtitle('X_R(e^{jw}) i.e. Real Part of X(e^{jw}) vs w');

for i = 1:4
    subplot(2, 2, i);
    plot(w, XRw{i});
    xlabel('w   (radians/sample)');
    ylabel(['X_R' ,'(e^{jw})']);
    ttl = {['[M = ', num2str(M(i)), ']']};
    if i > 2;  ttl = [{''}; ttl]; end;
    title(ttl);
    setDTFTradialAxis(1);
end
```



Plots above show that $X_R(e^{j\omega})$ has a sinc behaviour (as it has 2 sinc terms in its expression). These sincs have been shifted to $\pm 0.5\pi$. Hence, when used as a (digital) filter, this system will let through the CT frequencies,

$$\Omega = \frac{\omega}{T} = \frac{\pm\pi/2}{2\pi/\Omega_S} = \pm\frac{\Omega_S}{4}$$

and a few ones around it. Here, Ω_S is considered to be the sampling frequency of signal passed through the DT system $X(e^{j\omega})$.

As M is increased, $X_R(e^{j\omega})$ becomes more and more like sharp impulses at $\omega = \pm\pi/2$, and hence the frequencies other than $\pm\pi/2$ are more precisely eliminated.

Task 5

Statement:

The in-built $\mathrm{deconv}()$ function is useful in dividing two causal sequences. Write a MATLAB function $\mathrm{deconv\_m}()$ to divide two noncausal sequences, having the format:

$$\text{function}\,[p,\ \mathrm{np},\ r,\ \mathrm{nr}] = \mathrm{deconv\_m}(b,\ \mathrm{nb},\ a,\ \mathrm{na})$$

Check your function on the following operation:

$$\frac{z^2 + z + 1 + z^{-1} + z^{-2} + z^{-3}}{z + 2 + z^{-1}} \;=\; (z - 1 + 2z^{-1} - 2z^{-2}) \;+\; \frac{3z^{-2} + 3z^{-3}}{z + 2 + z^{-1}}$$

Code (*deconv_m.m*):

Code for the function $\mathrm{deconv\_m}()$, that can be used to divide (de-convolve) two noncausal sequences, is given below.

```
% <function>
% modified deconvolution routine for noncausal sequences
%
% <syntax>
% [p, np, r, nr] = deconv_m(b, nb, a, na)
%
% <I/O>
% b = numerator polynomial of support nb1 <= n <= nb2,       nb = [nb1, nb2]
% a = denominator polynomial of support na1 <= n <= na2,     na = [na1, na2]
% p = polynomial part of support np1 <= n <= np2,            np = [np1, np2]
% r = remainder part of support nr1 <= n <= nr2,             nr = [nr1, nr2]

function [p, np, r, nr] = deconv_m(b, nb, a, na)

    % perform deconvolution using inbuilt MATLAB function
    [p, r] = deconv(b, a);

    % compute n-axis values for quotient p
    np_b = nb(1) - na(1);                       % beginning n value for p
    np_e = np_b + length(p) - 1;                % ending n value for p
    np = np_b:np_e;

    % compute n-axis values for remainder r
    nr_b = nb(1);                               % beginning n value for r
    nr_e = nr_b + length(r) - 1;                % ending n value for r
    nr = nr_b:nr_e;

end
```

Testing $\text{deconv\_m}()$ function on given operation:

The code given below tests the above function $\text{deconv\_m}()$ using the z-domain expression provided in Task Statement, which can be treated as:

$$X_2(z) = \frac{X_3(z)}{X_1(z)} = \frac{z^2 + z + 1 + z^{-1} + z^{-2} + z^{-3}}{z + 2 + z^{-1}} = (z - 1 + 2z^{-1} - 2z^{-2}) + \frac{3z^{-2} + 3z^{-3}}{z + 2 + z^{-1}}$$

```
% z-inv of numerator (dividend) i.e. b[n] <==> X3(z)
b = [1 1 1 1 1 1];
nb = -2:3;

% z-inv of denominator (divisor) i.e. a[n] <==> X1(z)
a = [1 2 1];
na = -1:1;

% perform deconvolution on non-causal sequences (b[n] deconv a[n])
[p, np, r, nr] = deconv_m(b, nb, a, na)
```

```
p = 1×4
     1    -1     2    -2
np = 1×4
    -1     0     1     2
r = 1×6
     0     0     0     0     3     3
nr = 1×6
    -2    -1     0     1     2     3
```

```
% result in z-domain X2(z) = X3(z)/X1(z), using results p & r from deconv_m()
syms z;
X2 = (p*(z.^-np).') + (r*(z.^-nr).')/(a*(z.^-na).')
```

X2 =

$$z + \frac{\frac{3}{z^2} + \frac{3}{z^3}}{z + \frac{1}{z} + 2} + \frac{2}{z} - \frac{2}{z^2} - 1$$

```
% compare against known answer
X2_expected = (z - 1 + 2*z^-1 - 2*z^-2) + (3*z^-2 + 3*z^-3) / (z + 2 + z^-1);
if isequal(X2, X2_expected)
    disp(['The result of the deconv_m() function matches with the ',...
        'given answer for deconvolution (or z-domain division).'])
end
```

```
The result of the deconv_m() function matches with the given answer for deconvolution (or z-domain division).
```

The expression obtained from the results of $\text{deconv\_m}()$, given above by "X2", is the same as R.H.S. of $X_2(z)$ expression given at the top.

Task 6

Statement:

A stable system has four zeros and four poles as given below.

$$\text{Zeros}: \ \pm 1, \ \pm \text{j}1 \qquad\qquad \text{Poles}: \ \pm 0.9, \ \pm \text{j}0.9$$

It is also known that the frequency response function $H(e^{j\omega})$ evaluated at $\omega = \pi/4$ is equal to 1, that is,

$$H(e^{j\,\pi/4}) \ = \ 1$$

Question (1): Determine the system function $H(z)$, and indicate its region of convergence (ROC).

Question (2): Determine the difference equation representation.

Question (3): Determine the steady-state response $y_{\text{ss}}[n]$ if the input is $x[n] = \cos(\pi\, n/4)\, u[n]$.

Question (4): Determine the transient response $y_{\text{tr}}[n]$ if the input is $x[n] = \cos(\pi\, n/4)\, u[n]$.

Answers (1-4):

The work done on paper, i.e. answers to Q1, Q2, Q3 and Q4 above, starts from the next page.

The work done on paper starts from the next page.

Task 6

Given stable system with,

zeroes: $\pm 1, \pm j1$, poles: $\pm 0.9, \pm j0.9$
(4) (4)

and $H(e^{j\omega})\Big|_{\omega=\pi/4} = H(e^{j\pi/4}) = 1$

(Q 1)

System function $H(z)$ & its ROC:

from the zeroes and poles,

$$H(z) = K\frac{(z-1)(z+1)(z-j1)(z+j1)}{(z-0.9)(z+0.9)(z-j0.9)(z+j0.9)}$$

$$= K\frac{(z^2-1)(z^2-j^2)}{(z^2-0.81)(z^2-j^2 0.81)}$$

$$= K\frac{(z^2-1)(z^2+1)}{(z^2-0.81)(z^2+0.81)}$$

$$= K\frac{(z^4-1)}{(z^4-0.6561)}$$

$$H(z) = K \cdot \frac{1 - z^{-4}}{1 - 0.6561 z^{-4}}$$

put $z = e^{j\pi/4}$, $\left(H(e^{j\pi/4}) = 1\right)$

$$H(e^{j\pi/4}) = K \frac{1 - \left(e^{j\pi/4}\right)^{-4}}{1 - 0.6561\left(e^{j\pi/4}\right)^{-4}}$$

$$1 = K \frac{1 - e^{-j\pi}}{1 - 0.6561\, e^{-j\pi}}$$

$$1 = K \frac{1 - (-1)}{1 - 0.6561(-1)}$$

$$1 = K \frac{2}{1.6561}$$

$$K = \frac{1.6561}{2}$$

$$K = 0.82805$$

So, the system function $H(z)$ is,

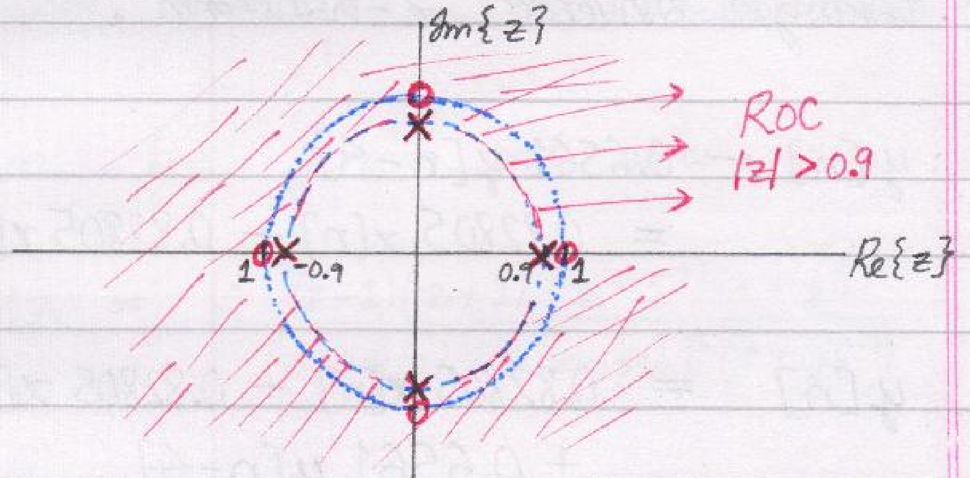
$$H(z) = 0.82805 \frac{1 - z^{-4}}{1 - 0.6561 z^{-4}}$$

— eq. (I)

for the ROC, consider that the system is stable (defined for $z = e^{jw}$ i.e. DTFT $H(e^{jw})$ exists), and since $z = \pm 0.9$ or $z = \pm j0.9$ is the outermost pole, so to have $|z| = 1$ in the ROC, the ROC must be,

$$|z| > 0.9$$

the pole-zero map is,



so the complete Z-domain representation of the system is,

$$H(z) = 0.82805 \frac{1 - z^{-4}}{1 - 0.6561 z^{-4}} \quad , \quad |z| > 0.9$$

— eq. (II)

(Q2)

Difference Equation representation:

$$H(z) = \frac{Y(z)}{X(z)} = 0.82805 \frac{1 - z^{-4}}{1 - 0.6561 z^{-4}}$$

$\Rightarrow$

$$Y(z) - 0.6561\, Y(z) z^{-4} = 0.82805\, X(z) - 0.82805\, X(z) z^{-4}$$

taking inverse z-transform,

$$y[n] - 0.6561\, y[n-4] = 0.82805\, x[n] - 0.82805\, x[n-4]$$

$$y[n] = 0.82805\, x[n] - 0.82805\, x[n-4] + 0.6561\, y[n-4]$$

— eq. (III)

(Q 3)

Steady-state response $y_{ss}[n]$:

for the input $x[n] = \cos\left(\frac{\pi n}{4}\right) u[n]$.

here, $\omega_0 = \pi/4$, so the Z-transform is,

$$X(z) = \frac{1 - \cos(\pi/4)\, z^{-1}}{1 - 2\cos(\pi/4)\, z^{-1} + z^{-2}} \quad (|z|>1)$$

$$= \frac{1 - 1/\sqrt{2}\, z^{-1}}{1 - 2(1/\sqrt{2})\, z^{-1} + z^{-2}}$$

$$= \frac{1 - 1/\sqrt{2}\, z^{-1}}{1 - \sqrt{2}\, z^{-1} + z^{-2}}, \quad |z|>1$$

or,

$$X(z) = \frac{1 - 1/\sqrt{2}\, z^{-1}}{1 - e^{j\pi/4} z^{-1} - e^{-j\pi/4} z^{-1} + z^{-2}}$$

$$= \frac{1 - 1/\sqrt{2}\, z^{-1}}{1(1 - e^{j\pi/4} z^{-1}) - e^{-j\pi/4} z^{-1}(1 - e^{j\pi/4} z^{-1})}$$

$$= \frac{1 - 1/\sqrt{2}\, z^{-1}}{(1 - e^{j\pi/4} z^{-1})(1 - e^{-j\pi/4} z^{-1})}$$

$(|z|>1)$ — eq. (IV)

So the system response is,

$$Y(z) = H(z)X(z) \quad , |z|>1 \qquad ROC=(|z|>0.9 \cap |z|>1)$$

$$= \frac{0.82805\,(1-z^{-4})}{(1-0.6561\,z^{-4})} \cdot \frac{(1-1/\sqrt{2}\,z^{-1})}{(1-\sqrt{2}\,z^{-1}+z^{-2})}$$

$$= \frac{0.82805\,(1-z^{-4})\,(1-1/\sqrt{2}\,z^{-1})}{(1-e^{j\pi/4}z^{-1})(1-e^{-j\pi/4}z^{-1})\,(1-0.9z^{-1})(1+0.9z^{-1}) \cdot (1+0.81z^{-2})}$$

— eq. (V)

[b] →

$$\left(= \frac{0.82805-0.414025\sqrt{2}\,z^{-1}-0.82805z^{-4}+0.414025\sqrt{2}z^{-5}}{1-2\sqrt{2}\,z^{-1}+z^{-2}-0.6561z^{-4}+0.6561\sqrt{2}z^{-5}-0.6561z^{-6}} \right)$$

[a] →

— eq. (VI)

continuing with eq. (V),

$$Y(z) = \frac{A}{1-e^{j\pi/4}z^{-1}} + \frac{B}{1-e^{-j\pi/4}z^{-1}} + \frac{C}{1-0.9z^{-1}} + \frac{D}{1+0.9z^{-1}} + \underbrace{\frac{Ez^{-1}+F}{1+0.81z^{-2}}}$$

$$= \frac{G}{1-j0.9z^{-1}} + \frac{H}{1+j0.9z^{-1}}$$

using $[r, p, k] = \text{residuez}(b, a)$ MATLAB function on the b, a coefficients of $Y(z)$, given by eq. (VI), we get the following partial fraction decomposition,

$$Y(z) = \frac{0.5}{1 - e^{j\pi/4} z^{-1}} + \frac{0.5}{1 - e^{-j\pi/4} z^{-1}} - \frac{0.0351}{1 - 0.9 z^{-1}} - \frac{0.0509}{1 + 0.9 z^{-1}} \underbrace{- \frac{0.0430 - j0.0755}{1 - j0.9 z^{-1}} - \frac{0.0430 + j0.0755}{1 + j0.9 z^{-1}}}_{= \frac{(-0.430 + j0.0755)(1 + j0.9 z^{-1}) + (-0.430 - j0.0755)(1 - j0.9 z^{-1})}{1 + 0.81 z^{-2}}}$$

so,

$$Y(z) = \frac{0.5}{1 - e^{j\pi/4} z^{-1}} + \frac{0.5}{1 - e^{-j\pi/4} z^{-1}} - \frac{0.0351}{1 - 0.9 z^{-1}} - \frac{0.0509}{1 + 0.9 z^{-1}} - \frac{0.086 + 0.1359 z^{-1}}{1 + 0.81 z^{-2}}$$

$$= \left.\frac{0.5}{1 - e^{j\pi/4} z^{-1}} + \frac{0.5}{1 - e^{-j\pi/4} z^{-1}}\right\} \text{poles at unit circle}$$

$$\underbrace{- \frac{0.0351}{1 - 0.9 z^{-1}} - \frac{0.0509}{1 + 0.9 z^{-1}} - \frac{0.086}{1 + 0.81 z^{-2}} - \frac{0.1359 z^{-1}}{1 + 0.81 z^{-2}}}_{\text{poles inside unit circle.}} \quad |z| > 1$$

— eq. (VII)

Now, the <u>Steady-state</u> response is due to poles that are at/on the unit circle.

In $Y(z)$ in eq. (VII), the first 2 terms correspond to poles at unit circle, so,

$$Y_{ss}(z) = \frac{0.5}{1-e^{j\pi/4}z^{-1}} + \frac{0.5}{1-e^{-j\pi/4}z^{-1}}$$

$$|z|>1$$

$\Rightarrow$ Taking inverse Z-transform,

$$y_{ss}[n] = 0.5\left(e^{j\frac{\pi}{4}}\right)^n u[n] + 0.5\left(e^{-j\pi/4}\right)^n u[n]$$

$$= \frac{1}{2}\left(e^{j\pi/4\, n} + e^{-j\pi/4\, n}\right) u[n]$$

$$y_{ss}[n] = \cos\left(\frac{\pi}{4} n\right) u[n] \quad \text{— eq. (VIII)}$$

So, eq. (VIII) gives the system's steady-state response. In this case, $y_{ss}[n] = y_p[n]$ (particular resp.), since these $Y_{ss}(z)$ terms also correspond to the input ($X(z)$) poles.

(Q 4)

Transient response $y_{tr}[n]$:

So, for the same input as in (Q3), $x[n] = \left(\cos\left(\frac{\pi}{4} n\right)\right) u[n]$, the output Z-transform $Y(z)$ is given by eq.(VII).

The transient response is due to poles that are inside the unit circle, which corresponds to the last 4 terms in $Y(z)$ in eq.(VII). So,

$$Y_{tr}(z) = -\frac{0.0351}{1-0.9z^{-1}} - \frac{0.0509}{1+0.9z^{-1}} - \frac{0.086}{1+0.81z^{-2}}$$

$$-\frac{0.1359z^{-1}}{1+0.81z^{-2}} \quad , \quad |z|>1$$

$$= -0.0351\frac{1}{1-(0.9)z^{-1}} - 0.0509\frac{1}{1-(-0.9)z^{-1}}$$

$$-0.086\frac{1-\cancel{\overset{a}{(0.9)}\overset{0}{(\cos \pi/2)} z^{-1}} = 0}{1-\cancel{2\underset{a}{(0.9)}\overset{0}{(\cos \pi/2)}} z^{-1} + \underset{a^2}{(0.9)^2} z^{-2}}$$

$$-0.151\frac{(0.9)\overset{1}{(\sin \pi/2)} z^{-1}}{1-2(0.9)\cancel{\underset{0}{(\cos \pi/2)}} z^{-1} + \underset{a^2}{(0.9)^2} z^{-2}}$$

$|z|>1$

Since the ROC is $|z|>1$, the sequences in $y_{tr}[n]$ must all be right-sided. So using the common Z-tr. pairs, the inverse Z-transform is,

$$
\begin{aligned}
y_{tr}[n] = & -0.0351\,(0.9)^n\,u[n] \\
& -0.0509\,(-0.9)^n\,u[n] \\
& -0.086\,(0.9)^n \cos\left(\tfrac{\pi}{2} n\right) u[n] \\
& -0.151\,(0.9)^n \sin\left(\tfrac{\pi}{2} n\right) u[n].
\end{aligned}
$$

— eq. (IX)

So, eq. (IX) gives the system's transient response. In this case, $y_{tr}[n] = y_h[n]$ (homogeneous resp.), since these $Y_{tr}(z)$ terms also correspond to the system ($H(z)$) poles.

So the answers to Q1-4 are:

(Q1)

$$H(z) = 0.82805\frac{1-z^{-4}}{1-0.6561z^{-4}} \qquad \text{ROC} = |z| > 0.9$$

(Q2)

$$y[n] = 0.82805\,x[n] - 0.82805\,x[n-4] + 0.6561\,y[n-4]$$

(Q3)

$$Y(z) = \frac{0.5}{1-e^{j\pi/4}z^{-1}} + \frac{0.5}{1-e^{-j\pi/4}z^{-1}} - \frac{0.0351}{1-0.9z^{-1}} - \frac{0.0509}{1+0.9z^{-1}} - \frac{0.086}{1+0.81z^{-2}} - \frac{0.1359z^{-1}}{1+0.81z^{-2}} \qquad \text{ROC} = |z| > 1$$

$$y_{\text{ss}}[n] = \cos\left(\frac{\pi}{4}n\right)\,u[n]$$

(Q4)

$$y_{\text{tr}}[n] = -0.0351(0.9)^n u[n] - 0.0509(-0.9)^n u[n] - 0.086(0.9)^n\cos\left(\frac{\pi}{2}n\right)u[n] - 0.151(0.9)^n\sin\left(\frac{\pi}{2}n\right)u[n]$$

Some parts of the computation, such as partial fraction decomposition and conformation of other results, were done in MATLAB with the following code.

Note: This code uses a modified version of the $\text{get\_zsys\_coeffs}()$ function implemented in Assignment 4.

```
% (Q1)
% ------------
% determining complete H(z)

zeros = [1 -1 1i -1i];                      % zeros in H(z)
poles = [0.9 -0.9 1i*0.9 -1i*0.9];          % poles in H(z)

% define H(z) using poles, zeros, and coefficient k
syms z k;
Hz = k * prod(z-zeros) / prod(z-poles);

% determine K, given H(e^jπ/4) = 1
Hw = subs(Hz, z, exp(1i*pi/4)) == 1;
K = vpa(solve(Hw, k))
```

K = 0.82805

```

% so complete H(z) is,
Hz = subs(Hz, k, K)
```

Hz =

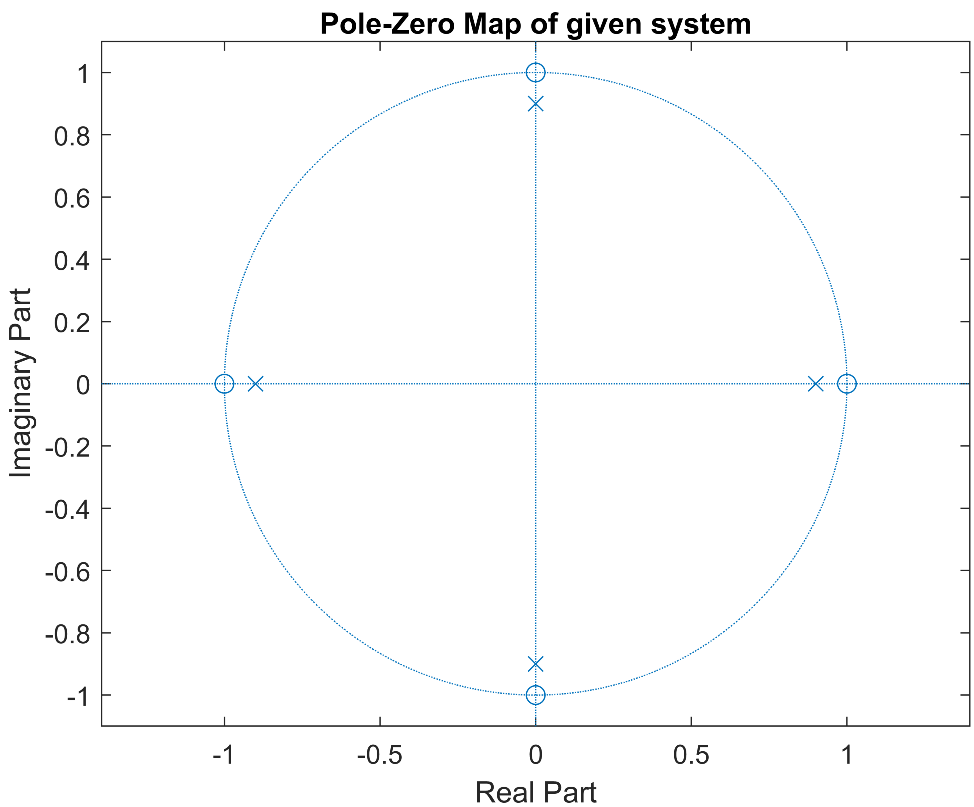
$$\frac{0.82805\,(z-1)\,(z+1)\,(z-\mathrm{i})\,(z+\mathrm{i})}{\left(z-\frac{9}{10}\right)\left(z+\frac{9}{10}\right)\left(z-\frac{9}{10}\,\mathrm{i}\right)\left(z+\frac{9}{10}\,\mathrm{i}\right)}$$

```
Hz = vpa(expand(Hz))
```

Hz =

$$\frac{0.82805\, z^4}{z^4 - 0.6561} - \frac{0.82805}{z^4 - 0.6561}$$

```
% pole-zero map
fig = figure;
zplane(zeros.', poles.');
title('Pole-Zero Map of given system');
```




```
% (Q3 & Q4)
% ------------
% determining the partial fraction form, and the inverse, of Y(z)

% define Y(z) = H(z)X(z) in terms of z^-1
% using paperwork equation on Page 28 (top)
syms z;
N = 0.82805*(1-z^-4)*(1-1/sqrt(2)*z^-1);          % numerator of Y(z)
D = ((1-0.6561*z^-4)*(1-sqrt(2)*z^-1+z^-2));      % denominator of Y(z)
Yz = N / D;
Yz = expand(Yz);

% find the CCDE coefficients of Y(z)
[b, a] = get_zsys_coeffs(N, D);
```

```
% check if the coefficients are correct against factorized Y(z)
Yz2 = sum(b.*z.^-(0:(length(b)-1)))/sum(a.*z.^-(0:(length(a)-1)));
Yz2 = expand(Yz2);
isequal(Yz, Yz2)
```

```
ans = logical
   1
```

```
% determine the partial fraction form of Y(z)
[r, p, k] = residuez(double(b), double(a))
```

```
r = 6×1 complex
   0.5000 + 0.0000i
   0.5000 - 0.0000i
  -0.0509 + 0.0000i
  -0.0430 + 0.0755i
  -0.0430 - 0.0755i
  -0.0351 + 0.0000i
p = 6×1 complex
   0.7071 + 0.7071i
   0.7071 - 0.7071i
  -0.9000 + 0.0000i
   0.0000 + 0.9000i
   0.0000 - 0.9000i
   0.9000 + 0.0000i
k =

     []
```

```
% display partial fraction sum symbolically
p = round(p, 4); r = round(r, 4); k = round(k, 4);
Yz = sum(r./(1-p*z^-1));
if ~isempty(k)
    Yz = Yz + sum(k.*z.^-(0:(length(k)-1)));
end
Yz = vpa(Yz, 5)
```

Yz =

$$\frac{0.0351}{\frac{0.9}{z}-1.0}-\frac{0.0509}{\frac{0.9}{z}+1.0}+\frac{0.043-0.0755\,\mathrm{i}}{-1.0+\frac{0.9\,\mathrm{i}}{z}}+\frac{-0.043-0.0755\,\mathrm{i}}{1.0+\frac{0.9\,\mathrm{i}}{z}}-\frac{0.5}{-1.0+\frac{0.7071-0.7071\,\mathrm{i}}{z}}-\frac{0.5}{-1.0+\frac{0.7071}{}}$$

The expression for $H(z)$ given by "Hz" in the code matches the expression derived on paper (Page 25, eq. (II)). The partial fraction decomposition of $Y(z)$ computed above ("Yz") has been used in paperwork (Page 29, eq. (VII)) to solve Q3 & Q4.