

University of Manchester
Department of Computer Science

First Semester Year 3 Examination Paper

CS3291: Digital Signal Processing

Date of Examination: January 2005

Answer THREE questions out of the five given.

Time allowed TWO HOURS

(Each question is marked out of 20). **Electronic calculators may be used.**

1. (a) Applications of digital signal processing (DSP) may be divided into two categories: 'real time' and 'non real time'. Explain these terms and give one example of a common application in each category. **[4 marks]**

(b) Why must real time DSP often be implemented in fixed point arithmetic and what special problems does this create for the programmer? **[4 marks]**

(c) The use of the MATLAB SP toolbox to design a 6th order low-pass FIR digital filter by the 'Remez exchange algorithm' produces the following matrix of coefficients:

$$[-0.0272 \quad 0.1250 \quad 0.2772 \quad 0.3537 \quad 0.2772 \quad 0.1250 \quad -0.0272]$$

Write a high-level language program to demonstrate how this digital filter may be implemented using integer arithmetic on a 16-bit fixed point DSP processor. **[6 marks]**

(d) Why is the Remez exchange algorithm generally considered to be superior to the windowing method as a design technique for FIR digital filters? **[3 marks]**

(e) Explain what is meant by the term 'linear phase' and why this is a desirable property for digital filters. State whether the 6th order FIR digital filter presented in part (c) above is exactly linear phase or only approximately so. **[3 marks]**

2.(a) An analogue signal $x_a(t)$ is band-limited to a frequency range below F Hz. This signal is sampled at f_s Hz to obtain the discrete time signal $\{x[n]\}$. Explain how it is possible, in principle, to reconstruct an exact replica of $x_a(t)$ from $\{x[n]\}$ provided $f_s > 2F$. **[8 marks]**

(b) Produce signal-flow-graphs for each of the following difference equations:

(i) $y[n] = x[n] - x[n - 2]$

(ii) $y[n] = x[n] - x[n - 1] - y[n - 1]$

For each of the difference equations above, determine its impulse-response and deduce whether the discrete-time system it represents is stable and/or causal. **[8 marks]**

(c) Calculate, by tabulation or otherwise, the output from difference equation (ii) in question 2 when the input is the impulse response of difference equation (i). **[4 marks]**

3. (a) With the aid of a block diagram, explain how analogue signals may be processed by digital signal processing techniques implemented on special purpose microprocessors. Briefly describe the function of each block in your diagram and indicate how its specification is affected by the bandwidths of the analogue input and output signals and the choice of sampling rate. **[10 marks]**

(b) Use the Fourier series approximation (windowing) method with a rectangular window to design a sixth order "low-pass" FIR digital filter whose cut-off frequency is one quarter of the sampling frequency and whose phase response is linear in the pass-band.

Give a signal flow graph for the digital filter.

How would you expect the gain and phase responses of this digital filter to be affected by (i) increasing the order and (ii) imposing a non-rectangular window? **[10 marks]**

4. Given that the system function of a third order Butterworth type analogue low-pass analogue filter with a 3 dB cut-off frequency of one radian/second is:

$$H(s) = \frac{1}{(1 + s)(1 + s + s^2)}$$

use the bilinear transformation to design a third order low-pass digital filter with a 3 dB cut-off frequency at one quarter of the sampling frequency. **[8 marks]**

Give a signal-flow-graph for the IIR filter. **[4 marks]**

With the aid of simple sketch graphs explain how frequency warping affects the frequency response of the digital filter. **[4 marks]**

Show how the digital filter would be implemented on a DSP microprocessor with floating point arithmetic. **[4 marks]**

- 5.(a) Define the following transforms and explain how they are related to each other:

- (i) Discrete time Fourier transform (DTFT)
- (ii) Discrete Fourier transform (DFT)
- (iii) Fast Fourier transform (FFT).

Explain how the DTFT is related to the analogue Fourier transform.

Explain why non-rectangular windows (Hann, etc) are generally preferred to the rectangular window when using the DFT for spectral estimation? **[10 marks]**

- (b) If the input signal to a digital filter with frequency response

$$H(e^{j\Omega}) = (5 + 2 \cos(\Omega)) e^{-j\Omega/2}$$

is $\{x[n]\}$ with $x[n] = 3 \cos(0.5n)$ for all n , what is the output signal?

[3 marks]

- (b) Give $H(z)$ for a DSP system with the following difference equation:

$$y[n] = x[n] + x[n-2] + 0.8 y[n-1]$$

Determine whether it is causal and stable and sketch its gain-response. **[7 marks]**

Solutions

1 (a)

Applications of digital signal processing can be divided into 'real time' and 'non real time' processing. A mobile phone contains a 'DSP' processor that is fast and powerful enough to perform the mathematical operations required to filter digitised speech (and process it in other ways as well) as the speech is being received. This is real time processing. [2]

A standard PC can perform 'non-real time' DSP processing on a stored recording of music and can take as much time as it needs to complete this processing. Non real time DSP is extremely useful in its own right; consider MP3 compression as an example. It is also used to 'simulate' the software for real time DSP systems before they are actually built into special purpose hardware, say for a mobile phone. The simulated DSP systems may be tested with stored segments of speech representative of what is expected when people talk into a mobile phone for real. [2]

(b)

Real time DSP is often implemented using 'fixed point' microprocessors since these consume much less power and are less expensive than 'floating point' devices. [1]

A fixed point processor deals with all numbers as though they were integers, and the numbers are often restricted to a word-length of only 16 bits. [1]

Overflow (numbers becoming too big for 16 bits) can easily occur with disastrous consequences to the sound quality. If we try to avoid the possibility of overflow by scaling numbers to keep them small in amplitude, they may then not be represented accurately enough as the quantisation (rounding or truncation to the nearest integer) incurs error which is a larger proportion of the value being represented. Fixed point DSP programming can be quite a difficult task. When we use a PC to perform non real time DSP, we normally have floating point operations available with word-lengths that are larger than 16 bits. This makes the task much easier. However when simulating fixed point real time DSP on a PC, it is possible to deliberately restrict the program to integer arithmetic, and this is very useful. [2]

(c)

Multiplying each coeff by a suitable power of two, e.g. 1024, and rounding to the nearest integer we obtain the vector A of integerised coeffs used below: [2]

```

K = 1024;
A = round([-0.027 0.125 0.277 0.354 0.277 0.125 -0.027 ]*K);
x = [0 0 0 0 0 0 0];
while 1
    x(1) = input('X = ');
    Y = A(1)*x(1);
    for k = 5 : -1 : 2
        Y = Y + A(k)*x(k);
        x(k) = x(k-1);
    end;
    Y = round(Y / K); % achieved by shifting
    disp([' Y = ' num2str(Y)]);
end;
```

Must mention that the division by K is achieved by shifting arithmetically right.

[3]
[1]

Question 1 continued

(d)

The Remez exchange algorithm gives an 'equi-ripple approximation' to the ideal gain response required; i.e. equal ripple peaks across pass-band and stop-band. [1]

With the windowing technique, the peaks of the stop-band ripples are not equal in amplitude and reduce with increasing frequency. The stop-band approximation gets better with increasing frequency. [1]

By making all ripple peaks equal, Remez minimises the difference between the ideal gain response and the approximation across the whole of the frequency range. It is a 'mini-max' approximation. Hence the highest stop-band ripple peak will be lower than for the windowing technique. [1]

(e)

Expressing the frequency-response $H(e^{j\Omega}) = G(\Omega)\exp(j\phi(\Omega))$ where $\phi(\Omega)$ is the phase-response, if $-\phi(\Omega)/\Omega = K$ which remains constant for all Ω in the range $-\pi$ to π , $H(e^{j\Omega})$ and the LTI system realising $H(e^{j\Omega})$ are said to be 'linear phase'. [1]

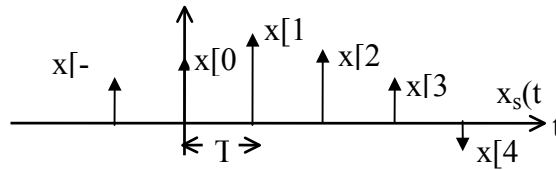
A linear phase system delays all frequency components of a periodic signal (say), expressed as a Fourier series say, by exactly the same amount of time, in this case K sampling intervals. If there are no amplitude changes, say because the signal falls within the pass-band of a low-pass digital filter, the output waveform will not be distorted in shape by phase effects (different frequency components being delayed by different amounts of time and therefore adding up differently). [1]

The given filter is exactly linear phase because the coeffs are symmetric about the third. [1]

2.(a)

Given $\{x[n]\}$ where $x[n]=x_a(nT)$ for all n .

Construct the impulse sequence (an analogue signal) $x_s(t)$ as sketched below:



[2]

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n] \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_a(t) \delta(t - nT) = \text{sample}_T \{ x_a(t) \}$$

By the Sampling Theorem, the Fourier transform of $x_s(t) = \text{'sample}_T \{ x_a(t) \}$ is

$$(1/T) \text{ repeat }_{2\pi/T} \{ X_a(j\omega) \}.$$

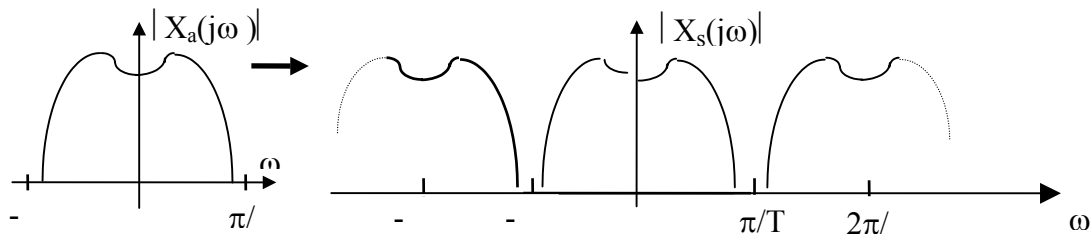
$$= X_s(j\omega) = \frac{1}{T} X_a(j\omega) + \frac{1}{T} X_a(j(\omega - 2\pi/T)) + \frac{1}{T} X_a(j(\omega + 2\pi/T)) + \dots$$

The spectrum of this analogue signal is "the sum of an infinite number of identical copies of $X_a(j\omega)$ each scaled by $1/T$ and shifted up or down in frequency by a multiple of $2\pi/T$ radians per second".

[2]

If $X_a(j\omega)$ is confined within $-\pi/T$ to π/T there is no overlap between frequency shifted copies of $X_a(j\omega)$ and we only need to remove the images from $X_s(j\omega)$ by filtering to leave a signal proportional to $x_a(t)$.

[2]

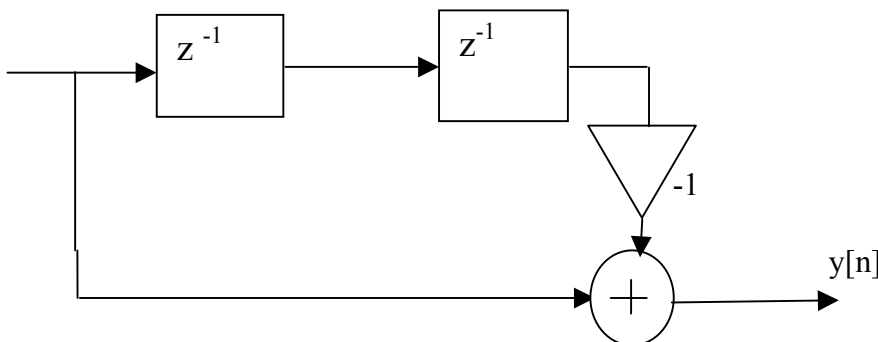


Therefore ideal reconstruction is to construct the impulse sequence $x_s(t)$ then filter it by an ideal low-pass filter with cut-off frequency π/T , i.e. half the sampling frequency, to remove the images. Then we need to scale by multiplying by the sampling interval T .

[2]

(b) Draw signal flow graphs.

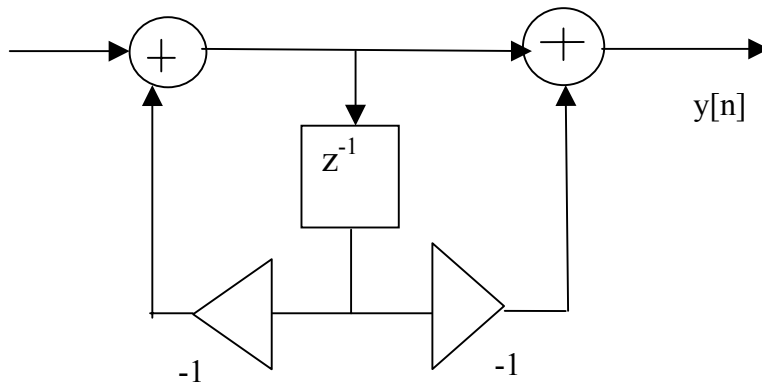
(i)



[1]

Question 2 continued

(ii)



[1]

(i) Impulse resp: $\{ \dots, 0, \dots, 0, \underline{1}, 0, -1, 0, \dots \}$
 It is causal and stable.

[1]

[1]

(ii)

n	x[n]	x[n-1]	y[n-1]	y[n]
0	1	0	0	1
1	0	1	1	-2
2	0	0	-2	2
3	0	0	-2	2
4	0	0	etc.....	

[2]

Impulse-response is: $\{ \dots, 0, \dots, 0, \underline{1}, -2, 2, -2, 2, -2, 2, \dots \}$
 It is causal but not stable.

[1]

[1]

(c)

n	x[n]	x[n-1]	y[n-1]	y[n]
0	1	0	0	1
1	0	1	1	-2
2	-1	0	-2	1
3	0	-1	1	0
4	0	0	0	0
5	0	0	0	0
			etc.....	

[3]

Output is:

$\{ \dots, 0, \dots, 0, \underline{1}, -2, 1, 0, 0, \dots \}$

[1]

This is perhaps surprisingly a finite sequence. An alternative method is:

$$X(z) = 1 - z^2$$

$$H(z) = \frac{1 - z^{-1}}{1 + z^{-1}}$$

Question 2 continued

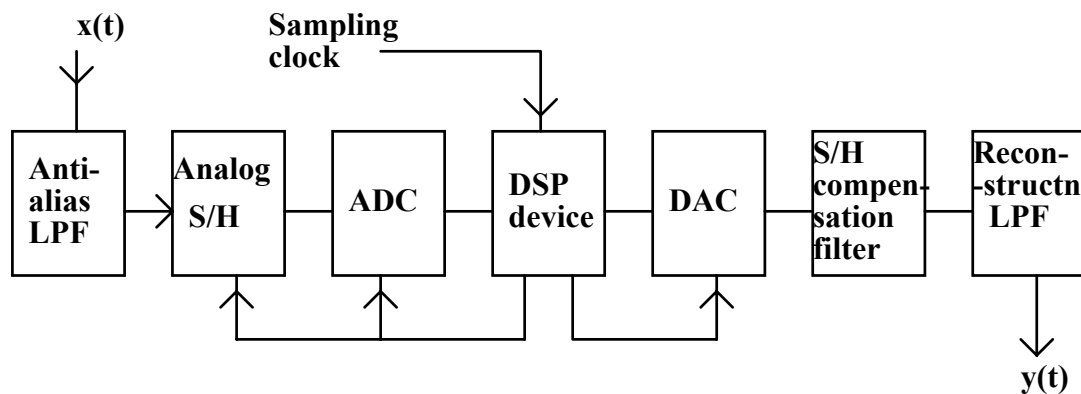
$$\begin{aligned} H(z)X(z) &= \frac{1-z^{-1}}{1+z^{-1}} \times (1-z^2) \\ &= \frac{(1-z^{-1})(1-z^{-1})(1+z^{-1})}{(1+z^{-1})} \\ (1-z^{-1})(1-z^{-1}) &= (1-2z^{-1}+z^{-2}) \end{aligned}$$

This is the z-transform of $\{ \dots, 0, \dots, 0, \underline{1}, -2, 1, 0, \dots, 0, \dots \}$

Therefore output is: $\{ \dots, 0, \dots, 0, \underline{1}, -2, 1, 0, \dots, 0, \dots \}$

3.(a)

Block diagram of a typical DSP system for processing analogue signals:



Antialiasing LPF: Analogue low-pass filter with cut-off frequency less than half the sampling frequency (f_s) to remove (strictly, to sufficiently attenuate) any spectral energy of the input signal $x(t)$ above $f_s/2$. When $x(t)$ is sampled, this spectral energy would otherwise produce, by aliasing, lower frequency energy capable of distorting the digitised input signal in the frequency range 0 to $f_s/2$. [1]

Analogue S/H: The analogue S/H circuit holds the input steady while the A/D conversion process takes place. [1]

ADC: Converts from analogue voltages to binary numbers of a specified wordlength. Quantisation error incurred. Samples taken at the "sampling frequency" f_s . [1]

DSP device: Digital processing system. Normally controls S/H and ADC to determine sampling rate which is normally fixed by a sampling clock connected via an input port to the processor. The processor reads samples from the ADC when they become available, processes them and outputs the resulting samples to the DAC. Many special-purpose DSP devices (microprocessors) have been designed specifically for this type of processing. [1]

DAC: Converts from binary numbers output by the processor to analogue voltages. "Zero order hold" or "stair-case like" waveforms are normally produced. [1]

S/H compensation: Zero order hold reconstruction multiplies the spectrum of the true output by $\text{sinc}(\pi f/f_s)$ which drops to about 0.64 at $f_s/2$. Hence we lose up to -4 dB. The S/H filter compensates for this effect by boosting the spectrum as it approaches $f_s/2$. Can be done digitally before the DAC or by an analogue filter after the DAC. [1]

Reconstruction LPF: Removes "images" of $-f_s/2$ to $f_s/2$ band produced by S/H reconstruction. Spec similar to that of input filter. [1]

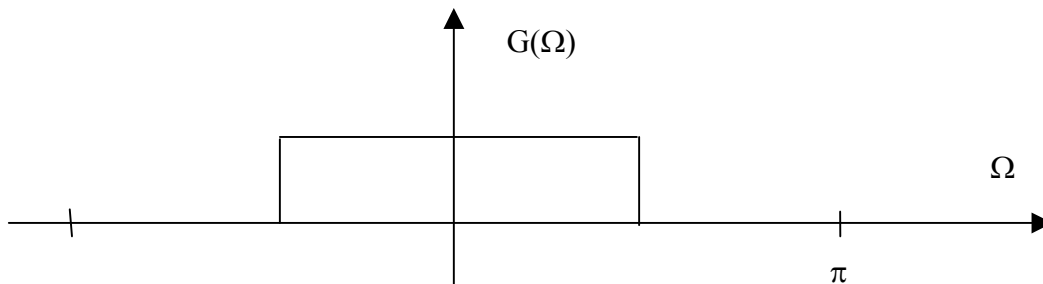
3(a) continued

Effect on detailed specification of input signal bandwidth & choice of sampling rate:

Input signal bandwidth determines lower bound for sampling rate f_s which must be greater than twice the input signal bandwidth. [1]

The closer f_s is to the theoretical minimum, however, the more critical will be the required characteristics of the analogue low-pass and S/H compensation filters. Increasing f_s has advantages in simplifying the requirements for these filters (their cut-off rates, for example) but the maximum possible value of f_s will ultimately be limited by the speed of the processor and the complexity of the required processing. [1]

Increasing the sampling rate also allows a reduction in the ADC and DAC wordlength to achieve the same signal-to-quantisation noise ratio, since the quantisation noise will be spread across a wider frequency domain, and some of it will be removed by the analogue reconstruction filter. About 3dB may be saved for each doubling of f_s , therefore multiplying f_s by 4 saves one bit. Bit-stream DAC techniques are based on this principle. [1]

3(b) Cut-off frequency = $f_s/4$ ie $\pi/2$ radians/sample.

[2]

Take phase to be zero initially.

Therefore $H(e^{j\Omega}) = G(\Omega)$

By inverse DTFT:

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(\Omega) e^{j\Omega n} d\Omega$$

$$= \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} e^{j\Omega n} d\Omega$$

[1]

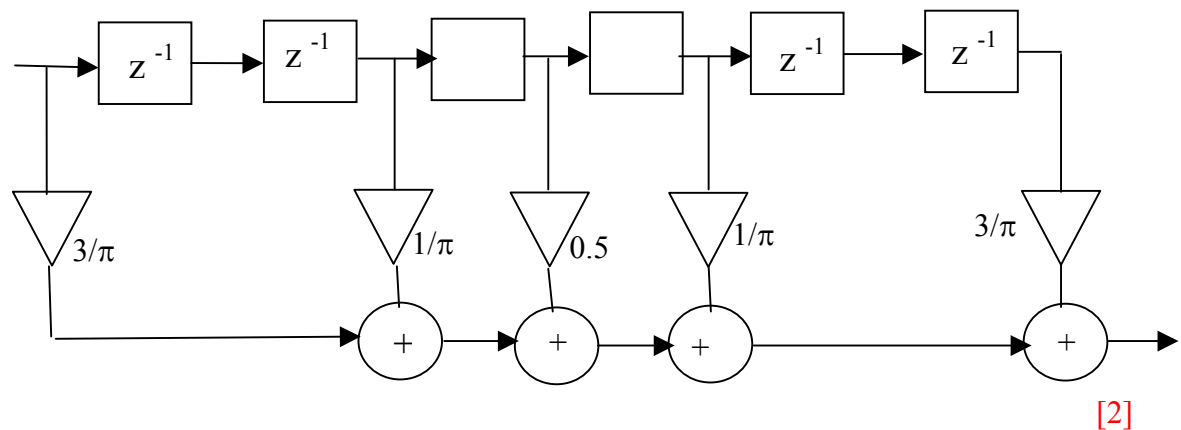
$$= \begin{cases} \frac{1}{2\pi} [\Omega]_{-\pi/2}^{\pi/2} & : n = 0 \\ \frac{1}{2\pi} \left[\frac{1}{jn} e^{j\Omega n} \right]_{-\pi/2}^{\pi/2} & n \neq 0 \end{cases}$$

$$= \begin{cases} 0.5 & : n = 0 \\ \frac{1}{2jn\pi} [e^{jn\pi/2} - e^{-jn\pi/2}] & : n \neq 0 \end{cases} = \begin{cases} 0.5 & : n = 0 \\ \frac{1}{n\pi} \sin(n\pi/2) & : n \neq 0 \end{cases} \quad [2]$$

$$\{h[n]\} = \{ \dots, 1/(5\pi), 0, -1/(3\pi), 0, 1/\pi, 0.5, 1/\pi, 0, -1/(3\pi), 1/(5\pi), 0, \dots \}$$

3(b) continued

Rectangularly windowing for -3 to $+3$ gives:



Filter is now linear phase with phase delay of 3 sampling intervals (in the pass-band) .

It will have a well defined stop-band decreasing in gain from 0dB at 0 Hz to -6 dB at the cut-off frequency. The stop-band gain will have ripples (illustration useful).

Increasing the order of the filter would mean that the phase delay would have to increase also if the filter remains linear phase. The magnitude response would become closer to the ideal low-pass response with more stop-band ripples. If the rectangular window is still used, the highest stop-band ripple would not reduce significantly due to Gibb's Phenomenon. [1]

The use of a Hann or similar raised cosine window would reduce the stop-band ripples at the expense of a less sharp cut-off rate from pass-band to stop-band. [1]

The phase response is not affected by the imposition of a non-rectangular window. [1]

4.

$$\Omega_c = \pi/4 \quad [1]$$

$$\begin{aligned} \text{Prewarped analogue cut-of frequency} &= 2 \tan(\Omega_c / 2) \\ &= 2 \text{ radians/second} \end{aligned} \quad [1]$$

Need to replace s by s/2 in expression for H(s).

$$H(s) = \frac{1}{(1+s/2+s^2/4)(1+s/2)} = \left(\frac{4}{4+2s+s^2} \right) \left(\frac{2}{2+s} \right) \quad [1]$$

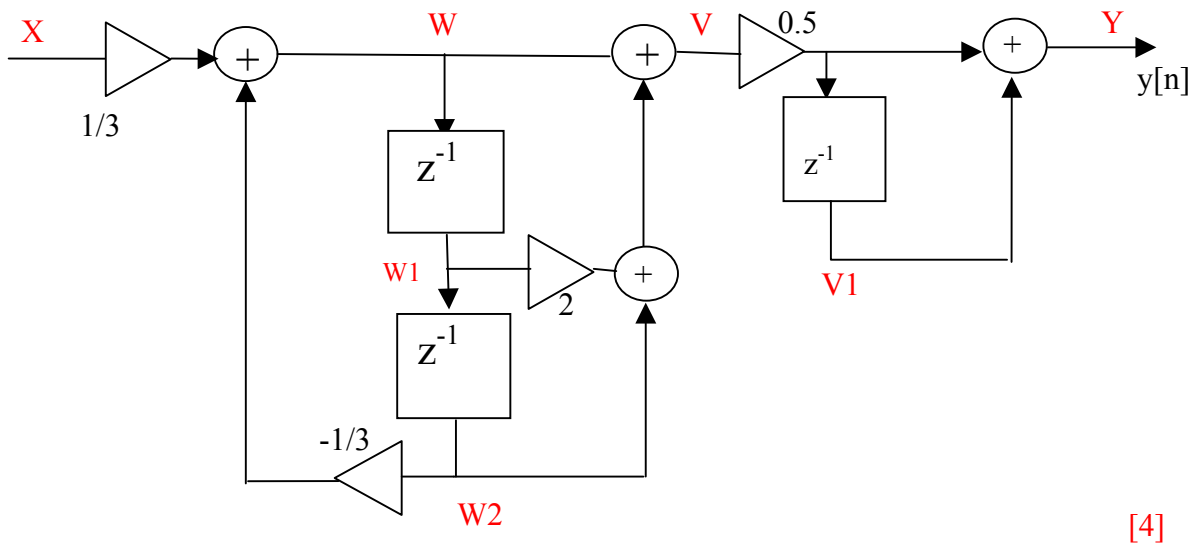
Bilinear transformation, taking T=1:

$$s \longrightarrow \frac{2(z-1)}{(z+1)} \quad [1]$$

$$H(s) = \left(\frac{4}{4 + \frac{4(z-1)}{(z+1)} + \frac{4(z-1)^2}{(z+1)^2}} \right) \left(\frac{2}{2 + \frac{2(z-1)}{(z+1)}} \right) \quad [1]$$

$$= \left(\frac{4(z+1)^2}{4(z+1)^2 + 4(z-1)(z+1) + 4(z-1)^2} \right) \left(\frac{2(z+1)}{2(z+1) + 2(z-1)} \right) \quad [1]$$

$$= \left(\frac{z^2 + 2z + 1}{3z^2 + 1} \right) \left(\frac{z+1}{2z} \right) = \left(\frac{1+2z^{-1}+1}{3+z^{-2}} \right) \left(\frac{1+z^{-1}}{2} \right) = \left(\frac{1}{3} \right) \left(\frac{1+2z^{-1}+1}{1+z^{-2}/3} \right) \left(\frac{1+z^{-1}}{2} \right) \quad [2]$$

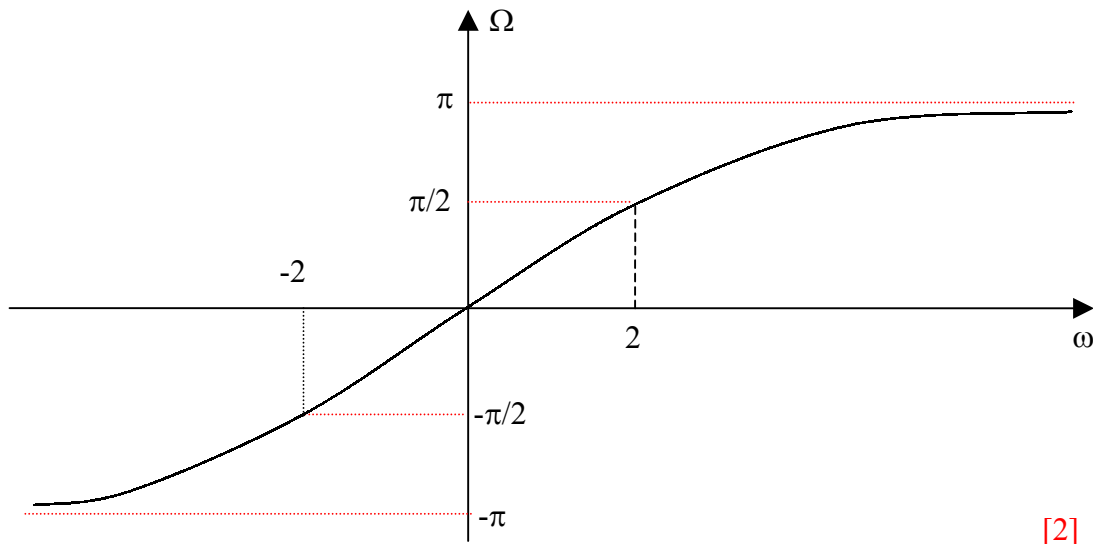


[4]

Question 4 continued

Effect of frequency warping

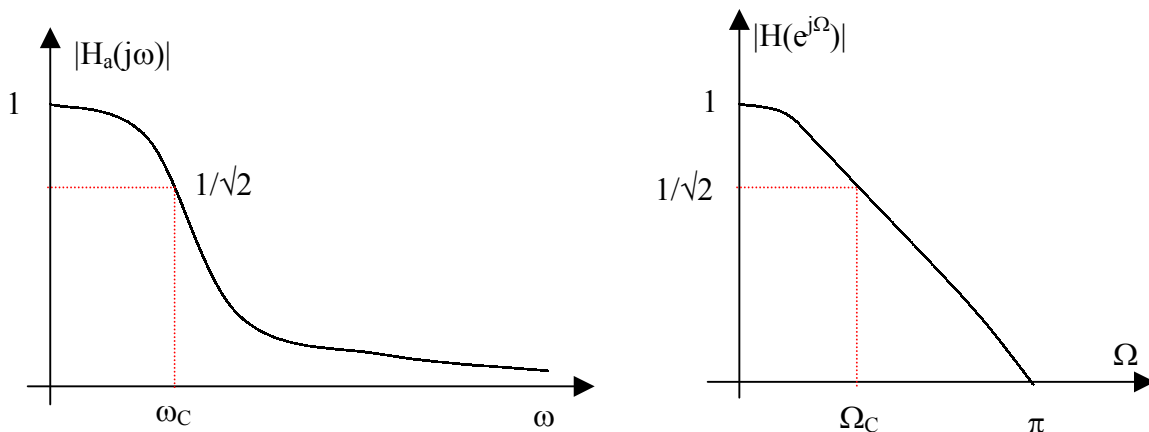
$$\Omega = 2 \tan^{-1}(\omega/2)$$



[2]

$H_a(j\omega)$ in the frequency range $-\infty < \omega < \infty$ radians / second is mapped to $H(e^{j\Omega})$ for Ω in the range $-\pi < \Omega < \pi$ radians/sample.

Mapping is reasonably linear for $-2 < \omega < 2$, but as $|\omega|$ increases from 2 towards infinity, the range of analogue frequencies mapped to a given range of values of Ω becomes larger and larger. Hence, in comparison to $|H_a(j\omega)|$, the discrete time version $|H(e^{j\Omega})|$ is changed in shape, as shown below for the low-pass filter.



Effect of frequency-warping makes the cut-off rate become greater as we approach $\Omega = \pi$.

[2]

Question 4 (continued)

```
W1=0; W2=0;
while 1
    X = input('X=');
    W = X / 3 - W2 / 3;
    V = W + 2*W1 + W2;
    W2 = W1;
    W1 = W;
    Y = 0.5*V + V1;
    V1 = V;
    disp(sprintf('output is: &f ', Y));
end;
```

[4]

5 (a)

DTFT

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \quad [1]$$

Transforms a discrete time signal $\{x[n]\}$ to the relative frequency domain where Ω is frequency in radians per sampling interval. It is discrete in time, but continuous in frequency and requires $x[n]$ to be defined for all values of n in the range $-\infty < n < \infty$ [2]

DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \quad [1]$$

Transforms a finite sequence $\{x[n]\}_{0,N-1}$ to the finite sequence $\{X[k]\}_{0,N-1}$.

For each k , $X[k] = X(e^{j\Omega_k})$ with $\Omega_k = 2\pi k/N$.

$X[k]$ is the complex valued sample of the spectrum $X(e^{j\Omega})$, where Ω is in the range 0 to 2π is uniformly sampled at $\Omega = 0, 2\pi/N, 2(2\pi/N), \dots, (N-1)(2\pi/N)$ [2]

FFT: A fast efficient algorithm for computing the DFT [1]

With $\Omega = \omega T$, $X(e^{j\Omega})$ is equal to the analogue Fourier transform, $X_s(j\omega)$, of a signal $x_s(t)$ comprising a sequence of impulses at intervals T seconds (the sampling period) each impulse weighted by the corresponding sample of $\{x[n]\}$. [2]

Non-rectangular windows reduce side-lobes and spectral spreading at the expense of spectral resolution. [2]

(b) $H(e^{j\Omega}) = (5 + 2 \cos(\Omega))e^{-j\Omega/2}$ and $\{x[n]\}$ with $x[n] = 3 \cos(0.5n)$

$$\begin{aligned} \text{When } \Omega = 0.5, H(e^{j\Omega}) &= (5 + 2 \cos(0.5))e^{-j0.25} \\ &= (5 + 2 \times 0.8776)e^{-j0.25} \\ &= 6.7552 e^{-j0.25} \end{aligned} \quad [1]$$

$$G(\Omega) = 6.7552 \quad \& \quad \phi(\Omega) = -0.25 \quad \text{when } \Omega = 0.5 \quad [1]$$

$$\begin{aligned} \text{Therefore output} &= 3 \times 6.7552 \cos(0.5n - 0.25) = 20.27 \cos(1.5n - 0.25) \\ \{y[n]\} &= \{20.27 \cos(1.5n - 0.25)\} \end{aligned} \quad [1]$$

Question 5 continued

(c)

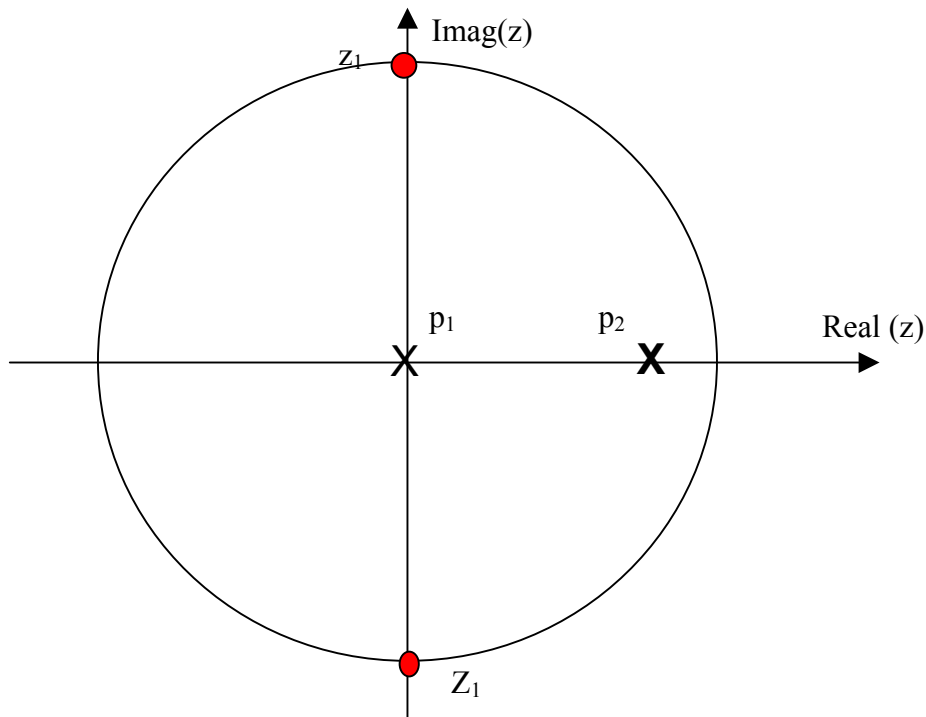
$$H(z) = \frac{1 + z^{-2}}{1 - 0.8z^{-1}}$$

$$= \frac{z^2 + 1}{z(z - 0.8)}$$

[1]

Zeros: $z_1 = +j$ & $z_2 = -j$ Poles: $p_1 = 0$, $p_2 = 0.8$

[1]



Filter is stable and causal because all poles lie inside the unit circle.

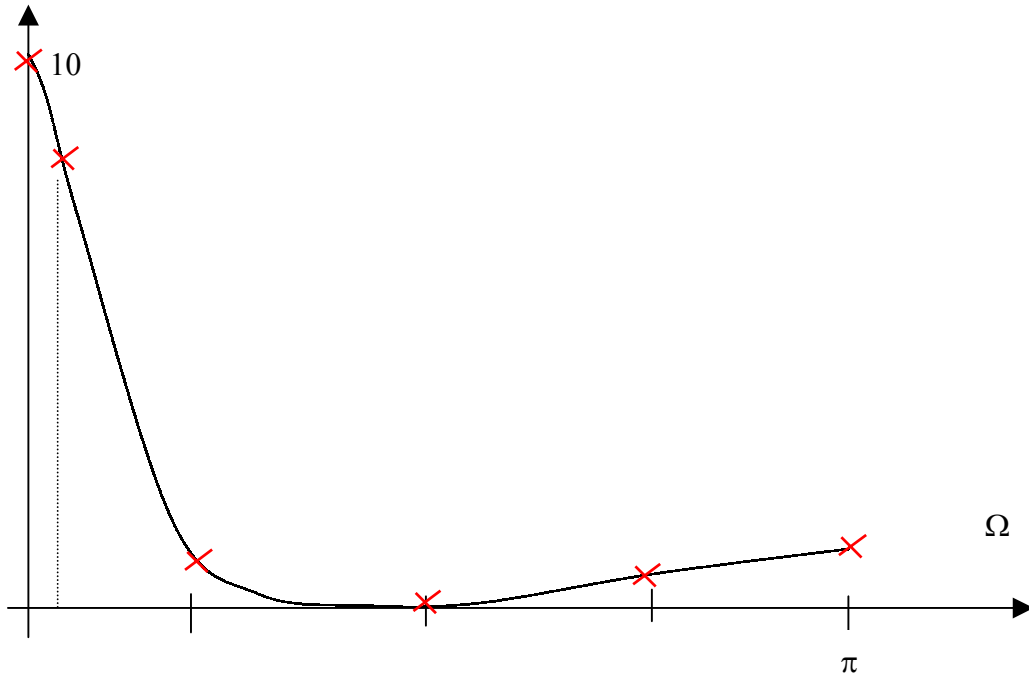
[1]

Estimation of gain response:

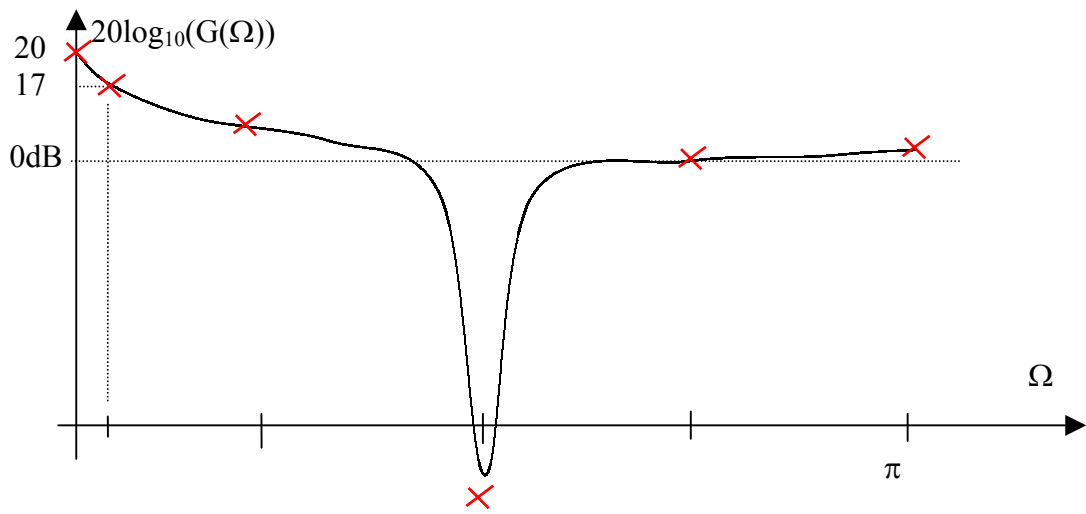
Ω	zero distances	pole distances	est gain	dB
0	1.4 x 1.4	1 x 0.2	10	20
$\pi/4$	0.7 x 1.7	1 x 0.7	1.7	5
$\pi/2$	0 x 2	1 x 1.2	0	-inf
$3\pi/4$	1.7 x 0.7	1 x 1.5	0.8	-0.2
π	1.4 x 1.4	1.8 x 1	1.2	2

[2]

3 dB pts: at 0.2, gain drops by approx 3 dB. to $10 / 1.4 = 7.1$



Alternatively, on a log scale:



[2]

Question 5:

Check (by Matlab)

```
freqz([1 0 1], [1 -.8]);
```

