

University of Manchester
CS3291: Digital Signal Processing 2003/2004

Solutions to selected problems in Notes

Section 1

1.1 Electrical waveform analogous to continuous variation in other quantity such as air pressure.

1.2. Never!

$$1.3. x(t) = \begin{cases} 0 & : t < 0 \\ \sin(100\pi t) & : t \geq 0 \end{cases}$$

1.4. Zero for $t < 0$.

1.5

$$x[n] = \begin{cases} 0 & : n < 0 \\ \sin(0.5\pi n) & : n \geq 0 \end{cases}$$

$$f_s = 10 \text{ Hz}$$

1.6. See notes.

1.7. (i) Multiplies by constant

(ii) Any linear time-invariant signal processing system of finite order.

(iii) Output is modulus of input (full wave...)

(iv) Multiplies given signal by another signal.

1.8. Filtering music to increase or decrease bass or treble power: HI-FI tone control.

1.9. Permanent storage of data: no ageing as with magnetic tape.

1.10. Loss of data can be catastrophic and total, e.g. due to a virus, rather than partial. Also data can be easily copied by unauthorized people.

1.11. Fixed point: only integers possible; fractions require a decimal or binary point to be assumed.

Floating point: fractions with wide dynamic range possible using mantissa and exponent.

1.12. The term "filter", is often assumed to be a device for removing or "filtering off" unwanted frequency components of a signal. However a filter may be more generally defined as any finite order LTI system. An analogue filter acts directly upon an analogue signal, for example using capacitors, inductors, and operational amplifiers. A digital filter operates to similar effect on digital signals.

1.13. The input signal is processed as it is being received sample by sample or block by block. Any output will be produced with fixed delay with respect to the input, as would be required, for example, for processing speech in a telephone conversation. The processor must be fast enough keep up with the incoming data. The opposite is batch mode processing where the incoming data may be captured and stored, for example, on a CD.

1.14. Restoring historical music recordings.

1.15. Analysing or predicting the behaviour of the stock exchange.

1.16. Using the "rectangular-to-polar" function on a calculator Modulus $(3+4j) = 5$, $\arg = 0.929$ radians.
Enter 3, shift or inv R->P, 4, "=", gives mod (5), "X<->Y gives arg (0.929)
Mod($-3e^{4j}$) = 3, Arg = $4+\pi$

1.17. $y(t) = -4\omega \sin(\omega t)$. Amplitude = 4ω for any ω .

1.18. $y(t) = (4/\omega) \sin(\omega t)$. Amplitude = $4/\omega$ for any $\omega > 0$

1.19. $(1 - e^{-(N+1)j\omega}) / (1 - e^{-j\omega})$ if $\omega > 0$ else $(N+1)$

1.20. $1 / (1 - 0.9e^{-j\omega})$

1.21. $1 + e^{-j\omega} = e^{-j\omega/2} (e^{j\omega/2} + e^{-j\omega/2}) = e^{-j\omega/2} (2\cos(\omega/2))$

1.22. $x^2 + 0.9x + 0.81 = (x - Re^{j\theta})(x - Re^{-j\theta})$. Therefore $R = 0.9$ & $2R\cos(\theta) = 0$; $\theta = \arccos(1/2) = 1.047$.

1.23. $x(t) = \cos(\omega t + \pi/2) + (1/2)\cos(2\omega t + \pi) + (1/3)\cos(3\omega t + 3\pi/2) + (1/4)\cos(4\omega t + 2\pi) + \dots$
 $= 0.5e^{j(\omega t + \pi/2)} + 0.5e^{-j(\omega t + \pi/2)} + 0.25e^{j(2\omega t + \pi)} + 0.25e^{-j(2\omega t + \pi)} + \dots$

1.24. Lecture notes.

1.25. Just look at formulae and observe what happens when ω is replaced by $-\omega$.

1.26. $G(0) = 1$ i.e. 0dB. $G(\omega_C) = 1/\sqrt{2}$ i.e. -3dB.

$G(10\omega_C) = 1/\sqrt{1+10^{2n}} \approx 1/\sqrt{10^{2n}} = 1/10^n$ i.e. -20n dB.

Section 2

2.1. Yes

2.2 If input to L_1 in top diagram is $\delta(t)$ output from L_1 is $h_1(t)$, i.e. its impulse-response, and this forms the input to L_2 to produce an overall output $h_1(t) \otimes h_2(t)$ as given by the convolution formula in notes. If input to L_2 in bottom diagram is $\delta(t)$ output from L_2 is $h_2(t)$ and this forms the input to L_1 to produce an overall output $h_2(t) \otimes h_1(t)$. Hence the impulse-response of the top arrangement is $h_1(t) \otimes h_2(t)$ and the impulse-response of the bottom arrangement is $h_2(t) \otimes h_1(t)$ and we know that $h_1(t) \otimes h_2(t) = h_2(t) \otimes h_1(t)$. So the two arrangements have exactly the same impulse-response and this means that their responses to any other input signal will also be identical. A similar argument may be used when L_1 and L_2 are discrete time LTI systems.

2.3.

$$H(j\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt = \int_0^1 e^{-j\omega t} dt = \begin{cases} \frac{1}{-j\omega} [e^{-j\omega t}]_0^1 & : \omega \neq 0 \\ 1 & : \omega = 0 \end{cases}$$

$$= \frac{e^{-j\omega/2}}{-j\omega} [e^{-j\omega/2} - e^{j\omega/2}] = (2/\omega)e^{-j\omega/2} \sin(\omega/2) \quad \text{when } \omega \neq 0$$

$$= e^{-j\omega/2} \text{sinc}(\omega/(2\pi)) \quad \text{for any } \omega$$

2.4. $H(j\omega) = e^{-3j\omega/2} \text{sinc}(\omega/(2\pi))$ for any ω .

Same gain-response, but different phase-response.

2.5. System would be stable and causal.

Calculate Fourier transform of $h(t)$.

2.6.

$$|H(j\omega)| = \left| \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt \right| \leq \int_{-\infty}^{\infty} |h(t)e^{-j\omega t}| dt = \int_{-\infty}^{\infty} |h(t)| dt = \text{finite}$$

2.7. By applying the inverse Fourier transform and performing a calculation similar to that in 2.3, we get $h(t) = (1/\pi) \text{sinc}(t/\pi)$. This is a 'sinc' function of time that remains non-zero for all t from $-\infty$ to $+\infty$ and hence is the impulse-response of a non-causal filter.

Section 3

3.1. We can find two simple signals for which the rule for linearity does not apply.

Let $x_1[n]=1$ for all n and $x_2[n]=-1$ for all n .

Response to $\{x_1[n]\}$ is $\{y_1[n]\}$ with $y_1[n]=1$ for all n .

Response to $\{x_2[n]\}$ is $\{y_2[n]\}$ with $y_2[n]=1$ for all n .

As $x_1[n]+x_2[n]=0$ for all n , the response to $\{x_1[n]+x_2[n]\}$ will be $\{0^2\}$ i.e. zero for all n .

But this is not $\{y_1[n]+y_2[n]\}$ which would be 2 for all n .

3.2. $\{ \dots, 0, \dots, \underline{1}, 1, 1, 1, -4, 0, \dots, 0, \dots \}$

3.3. Signal flow graph (i) is non-recursive like fig 3.3 whereas (ii) is recursive like fig 3.4.

3.4. (i) $\{ \dots, 0, \dots, \underline{1}, 1, 0, \dots, 0, \dots \}$ stable & causal. It is a finite impulse response.

(ii) $\{ \dots, 0, \dots, \underline{1}, -1, 1, -1, 1, -1, \dots, 1, -1, \dots \}$ causal but unstable. It is an infinite impulse response.

Note that not all filters with infinite impulse responses are unstable, but this one is.

3.5. $\{ \dots, 0, \dots, 0, \underline{1}, 0, 0, \dots, 0, \dots \}$

3.6. 800 Hz.

3.7. $H(e^{j\Omega}) = 1 - e^{-j\Omega} = e^{-j\Omega/2} (e^{j\Omega/2} - e^{-j\Omega/2}) = 2j \sin(\Omega/2) e^{-j\Omega/2} = 2 \sin(\Omega/2) e^{-j(\Omega/2 - \pi/2)}$

$$G(\Omega) = |2 \sin(\Omega/2)|$$

When $\Omega > 0$ then $\phi(\Omega) = -(\Omega/2 - \pi/2)$; when $\Omega < 0$ then $\phi(\Omega) = \pi - (\Omega/2 - \pi/2) = -(\Omega/2 + \pi/2)$.

When $\Omega = 0$ phase is arbitrary as $G(\Omega) = 0$, call it zero.

This is not linear phase.

3.8. $\{ \dots, 0, \dots, 0, \underline{4}, -8, 16, -32, \dots \}$ Non-stable.

3.9. We know that $\{e^{j\Omega n}\}$ produces $\{H(e^{j\Omega}) e^{j\Omega n}\}$ and therefore $\{e^{-j\Omega n}\}$ produces $\{H(e^{-j\Omega}) e^{-j\Omega n}\}$

We also know that $H(e^{-j\Omega})$ is the complex conjugate of $H(e^{j\Omega})$.

So that since $H(e^{j\Omega}) = G(\Omega)e^{j\phi(\Omega)}$ it follows that $H(e^{-j\Omega}) = G(\Omega)e^{-j\phi(\Omega)}$.

Now $\{\cos(\Omega n)\} = \{0.5 (e^{j\Omega n} + e^{-j\Omega n})\} = 0.5 \{e^{j\Omega n}\} + 0.5 \{e^{-j\Omega n}\}$.

Therefore the response to $\{\cos(\Omega n)\}$ is $0.5 \{G(\Omega)e^{j\phi(\Omega)}e^{j\Omega n}\} + 0.5 \{G(\Omega)e^{-j\phi(\Omega)}e^{-j\Omega n}\}$
 $= 0.5 G(\Omega) \{e^{j(\phi(\Omega)+\Omega n)} + e^{-j(\phi(\Omega)+\Omega n)}\} = \{G(\Omega) \cos(\Omega n + \phi(\Omega))\}$

$$3.10. H(e^{j\Omega}) = 1 + 2e^{-j\Omega} + 3e^{-2j\Omega} + 2e^{-3j\Omega} + e^{-4j\Omega} = e^{-2j\Omega} (e^{2j\Omega} + 2e^{j\Omega} + 3 + 2e^{-j\Omega} + e^{-2j\Omega})$$

$$= e^{-2j\Omega} (3 + 4\cos(\Omega) + 2\cos(2\Omega))$$

It may be shown by various means that $3 + 4\cos(\Omega) + 2\cos(2\Omega) \geq 0$ for all Ω .

(For example show that $3 + 4\cos(\Omega) + 2\cos(2\Omega) = 1 + 4\cos(\Omega) + 4\cos^2(\Omega) = (1 + 2\cos(\Omega))^2$)

Therefore the phase response $\phi(\Omega)$ is -2Ω for all Ω .

This is linear phase with a phase delay of 2 samples.

Note that the impulse response $\{\dots, 0, \dots, 0, \underline{1}, 2, 3, 2, 1, 0, \dots, 0, \dots\}$ is symmetric about $n=2$.

A symmetric impulse response gives a linear phase response.

Section 4

4.1. Cut-off frequency = $\pi/2$ radians/sample.

$$G(\Omega) = \begin{cases} 1 & -\pi/2 \leq \Omega \leq \pi/2 \\ 0 & \pi/2 < |\Omega| < \pi \end{cases}$$

Taking $\phi(\Omega)$ to be 0, $H(e^{j\Omega}) = G(\Omega)$ and the ideal impulse response is, by the inverse DTFT:

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{j\Omega n} d\Omega = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} e^{j\Omega n} d\Omega = \frac{\sin(n\pi/2)}{n\pi} \text{ when } n \neq 0$$

$$h[0] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{j\Omega n} d\Omega = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} e^{j\Omega n} d\Omega = \frac{\sin(n\pi/2)}{n\pi} \text{ when } n \neq 0$$

It may be checked that $h[n] = 0.5$ when $n=0$

Therefore $\{h[n]\}$ is the following infinite sequence:

$$\{\dots, -1/(7\pi), 0, 1/(5\pi), 0, -1/(3\pi), 0, 1/\pi, \underline{0.5}, 1/\pi, 0, -1/(3\pi), 0, 1/(5\pi), 0, -1/(7\pi), \dots\}$$

Rectangularly windowing for $-5 \leq n \leq 5$ gives the following sequence:

$$\{\dots, 0, \dots, 0, 1/(5\pi), 0, -1/(3\pi), 0, 1/\pi, \underline{0.5}, 1/\pi, 0, -1/(3\pi), 0, 1/(5\pi), 0, \dots, 0, \dots\}$$

Delaying by 5 samples to make the impulse response causal gives:

$$\{\dots, 0, \dots, 0, \underline{1/(5\pi)}, 0, -1/(3\pi), 0, 1/\pi, 0.5, 1/\pi, 0, -1/(3\pi), 0, 1/(5\pi), 0, \dots, 0, \dots\}$$

We can now draw the signal-flow graph of the FIR filter.

$$H(z) = 1/(5\pi) - (1/3\pi)z^{-2} + (1/\pi)z^{-4} - 0.5z^{-5} + (1/\pi)z^{-6} - (1/3\pi)z^{-8} + (1/5\pi)z^{-10}$$

$$y[n] = (1/(5\pi))x[n] - (1/3\pi)x[n-2] + (1/\pi)x[n-4] - 0.5x[n-5] + (1/\pi)x[n-6] - (1/3\pi)x[n-8] + (1/5\pi)x[n-10]$$

4.2.

$$G(\Omega) = \begin{cases} 0 & \pi/2 < |\Omega| < \pi \\ 1 & -\pi/4 < \Omega < \pi/4 \\ 0 & -\pi/4 < \Omega < \pi/4 \end{cases}$$

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{j\Omega n} d\Omega = \frac{1}{2\pi} \int_{-\pi/4}^{\pi/4} e^{j\Omega n} d\Omega + \frac{1}{2\pi} \int_{\pi/4}^{\pi/2} e^{j\Omega n} d\Omega$$

= $(1/n\pi)(\sin(n\pi/2) - \sin(n\pi/4))$ when $n \neq 0$ and 0.25 when $n=0$. Hence etc.

4.3.

$$y[n] = (1/(5\pi))x[n] - (1/(3\pi))x[n-2] + (1/\pi)x[n-4] + 0.5x[n-5] + (1/\pi)x[n-6] - (1/(3\pi))x[n-8] + (1/(5\pi))x[n-10]$$

$$= 0.06366x[n] - 0.1061x[n-2] + 0.3183x[n-4] + 0.5x[n-5] + 0.3183x[n-6] - 0.1061x[n-8] + 0.06366x[n-10]$$

$$= \frac{637x[n] - 1061x[n-2] + 3183x[n-4] + 5000x[n-5] + 3183x[n-6] - 1061x[n-8] + 637x[n-10]}{10000}$$

The following is the “bare bones” of a C program to read 1000 16-bit integer samples of a signal from a binary file, pass these through the 10th order filter above implemented using integer arithmetic only, and store the output samples generated in a binary output file.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
void main( )
{ file *fpin;fpout;
  long n, y;
  short m,ix, iy, x[11], a[11]; //16-bit integers
  fpin=fopen(c:\\.\\infilename.dat, “rb”);
  fpout=fopen(c:\\.\\outfilename.dat, “wb”);
  a[0]=637; a[1]=0; a[2]=-1061; a[3]=0; a[4]=3183; etc.
  for (m=1;m<11;n++) x[m]=0;
  for ( n=0;n<1000;n++)
  { fread(&ix, sizeof(short),1, fpin); x[0] = ix;
    y=x[0]*a[0];
    for (m=10; m>0; m--) {y = y + x[m]*a[m]; x[m] = x[m-1];}
    iy = y / 10000;
    fwrite(&iy, sizeof(short),1, fpout); }
  fclose(fpin); fclose(fpout);
}
```

4.4.

$$G(\Omega) = \begin{cases} 1 & -\pi < \Omega < -\pi/3 \\ 0 & -\pi/3 \leq \Omega \leq \pi/3 \\ 1 & \pi/3 < \Omega < \pi \\ 0 & \pi \leq \Omega < 2\pi \end{cases}$$

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{j\Omega n} d\Omega = \frac{1}{2\pi} \int_{-\pi}^{-\pi/3} e^{j\Omega n} d\Omega + \frac{1}{2\pi} \int_{\pi/3}^{\pi} e^{j\Omega n} d\Omega$$

$$h[n] \equiv \frac{1}{2\pi} \int_{-\pi}^{-\pi/3} (e^{j\Omega n}) d\Omega + \frac{1}{2\pi} \int_{\pi/3}^{\pi} (e^{j\Omega n}) d\Omega = \frac{1}{2\pi} \int_{-\pi}^{-\pi/3} e^{j\Omega n} d\Omega + \frac{1}{2\pi} \int_{\pi/3}^{\pi} e^{j\Omega n} d\Omega$$

4.5. no

4.6. Straightforward calculation.

4.7. If filter is linear phase with phase-delay N , then $-\phi(\Omega)/\Omega = N$ sampling intervals. Therefore:

$$h[n] = (1/\pi) \int_0^\pi |H(e^{j\Omega})| \cos(\Omega(n-N)) d\Omega$$

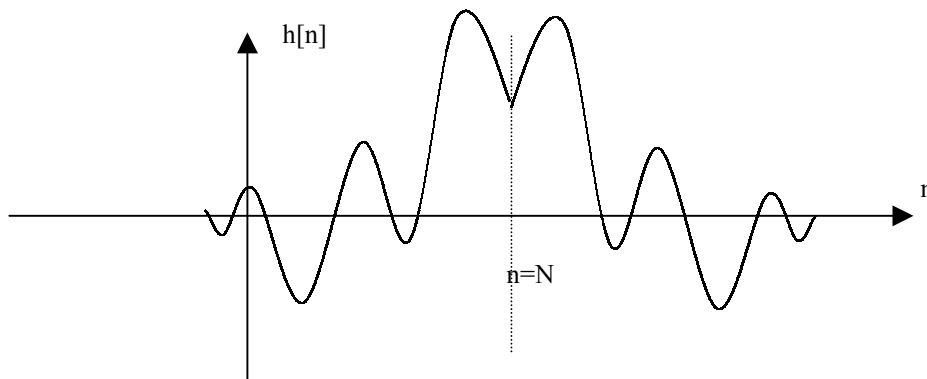
Now

$$h[N+n] = (1/\pi) \int_0^\pi |H(e^{j\Omega})| \cos(\Omega n) d\Omega$$

$$h[N-n] = (1/\pi) \int_0^\pi |H(e^{j\Omega})| \cos(\Omega(-n)) d\Omega$$

$$= (1/\pi) \int_0^\pi |H(e^{j\Omega})| \cos(\Omega n) d\Omega = h[N+n]$$

since $\cos(-\theta) = \cos(\theta)$ for any θ .



So $h[n]$ must be symmetric about $n=N$ as sketched above. If $h\{[n]\}$ is an infinite impulse-response, we will clearly have a problem with causality since if it goes forward for all time, it must also go backward for all time and give us non-zero values of $h[n]$ for $n < 0$. We cannot have an IIR digital filter which is exactly linear phase.

But we can have an FIR digital filter which is exactly linear phase. Can you see why?

Note: By a similar argument, a linear phase analogue filter must have an impulse-response which is symmetric about some point in time, $t=D$ say. This means that for all t , $h(D+t) = h(D-t)$, and if $h(t)$ remains non-zero as $t \rightarrow \infty$, $h(t)$ must also remain non-zero as t tends to $-\infty$. But analogue filters have infinite impulse-responses which means that an exactly linear phase analogue filter must be non-causal and therefore unrealisable.

The argument is the same for discrete time filters where $h[n]$ is symmetric about $n=N$ where N is an integer; i.e. $h[N-n]=h[N+n]$ for all n .

It is also easily shown to be true where $2N$ is an integer and $h[N+0.5+n]=h[N-0.5-n]$ for all n .

The argument for discrete time filters is a little more complicated where the symmetry is about $n=M$ and $2M$ is not an integer. Fortunately we rarely encounter this case.

Examples of symmetric impulse-responses corresponding to linear phase FIR digital filters are:

$$\{\dots, 0, \dots, 0, \underline{1}, -2, 3, 7, 3, -2, 1, 0, \dots, 0, \dots\} \quad N=3 \quad ; \quad x[2] = x[4], \text{ etc.}$$

$$\{\dots, 0, \dots, 0, \underline{1}, -2, 5, 5, -2, 1, 0, \dots, 0, \dots\} \quad N=2.5: \quad x[2]=x[3], x[1]=x[4], x[0]=x[5]$$

Section 5

5.1. (a) $H(z) = 2 - 3z^{-1} + 6z^{-4}$
 (b) $H(z) = z^{-1} / (1 + z^{-1} + 0.5z^{-2})$

5.2. Corresponding difference equation is $y[n] = x[n-1]$

5.3. (i) When input $x[n] = z^n$ then output $y[n] = H(z) z^n$.

$$\text{Substituting, } H(z)z^n = z^n - 0.9 H(z)z^{n-1} \quad \text{Therefore } H(z) [z^n + 0.9z^{n-1}] = z^n$$

$$H(z) = 1 / (1 + 0.9z^{-1})$$

(ii) $\{h[n]\} = \{ \dots, 0, \dots, 0, \underline{1}, -0.9, 0.81, -0.9^3, 0.9^4, \dots \}$

$$H(z) = 1 + (-0.9z^{-1}) + (-0.9)^2 z^{-2} + (-0.9)^3 z^{-3} + \dots$$

$$= 1 + (-0.9z^{-1}) + (-0.9z^{-1})^2 + (-0.9z^{-1})^3 + \dots$$

$$= 1 / (1 + 0.9z^{-1}) \text{ assuming } |0.9z^{-1}| < 1 \text{ i.e. } |z| > 0.9.$$

5.4. $y[n] = x[n] + 3x[n-1] + 2x[n-2] - 0.9 y[n-1]$

$$\text{Zeros: } z = -2 \text{ \& } z = -1; \quad \text{poles at } z = 0 \text{ \& } z = -0.9.$$

5.5. Difference equation is: $y[n] = x[n] + 2 y[n-1]$.

$$\{h[n]\} = \{ \dots, 0, \dots, 0, 1, 2, 4, 8, 16, 32, \dots \} \quad \text{Unstable! (NB not all IIR filters are unstable)}$$

5.6.

$$H(z) = \frac{1 - 0.9z^{-1} + 0.81z^{-2}}{1 - 0.95z^{-1} + 0.9025z^{-2}} = \frac{z^2 - 0.9z + 0.81}{z^2 - 0.95z + 0.9025} = \frac{(z - 0.9e^{j\pi/3})(z - 0.9e^{-j\pi/3})}{(z - 0.95e^{j\pi/3})(z - 0.95e^{-j\pi/3})}$$

The gain response will have a peak of amplitude 2 (6dB) at $\Omega = \pi/3$. The gain at frequencies not close to $\pi/3$ will be approximately one. To find out how sharp the peak is we can do various easy things.

You may choose to estimate the gain at $\pi/3 \pm 0.05$. A bit of geometry (right-angle triangles as usual) tells us that the gain at $\pi/3 \pm 0.05$ is $\sqrt{(0.1^2 + 0.05^2)} / \sqrt{(0.05^2 + 0.05^2)} = \sqrt{5} / \sqrt{2} = 4$ dB. The distance to the pole has increased by a factor $\sqrt{2}$ (decreasing the gain by 3dB) but the distance to the zero has increased from 0.1 to 0.112, i.e. a factor 1.12 (corresponding to an increase in gain of about 1dB).

Overall the gain decreases by 2 dB from its value of 6 dB at $\pi/3$. This information will allow a reasonable sketch (showing 4 dB points rather than 3 dB points), but if you insist on finding the 3dB points, you can do it quite easily by finding the increase θ in relative frequency such that (distance to zero) / (distance to pole)

reduces from 2 to approximately $\sqrt{2}$. As usual we neglect changes to the distances to the complex conjugates of this pole and zero as they are far away.

$$\text{Distance to zero} \approx \sqrt{(\theta^2 + 0.1^2)} \text{ and distance to pole} \approx \sqrt{(\theta^2 + 0.05^2)}.$$

$$\text{Solving } \sqrt{(\theta^2 + 0.1^2)} / \sqrt{(\theta^2 + 0.05^2)} = \sqrt{2} \text{ gives } \theta = \pm 0.071 \text{ radians/sample.}$$

Hence 3dB points are at $\pi/3 \pm 0.071$.

Follow-up exercise: repeat this problem with the poles at $0.99\exp(\pm j\pi/3)$ and the zeros unchanged.

The peak at $\pi/3$ now becomes much higher, i.e. 20 dB, and you will find that the points where the gain drops by 3 dB from 20 dB to 17 dB occur very close to $\pi/3 \pm 0.01$. But where does the gain become approximately 3dB? Solving $\sqrt{(\theta^2 + 0.1^2)} / \sqrt{(\theta^2 + 0.01^2)} = \sqrt{2}$ gives $\theta = \pm 0.1$ radians/sample; i.e. the gain drops to 3dB at $\Omega = \pi/3 \pm 0.1$. A gain response sketch for this follow-up exercise can therefore be drawn with minimal calculation.

5.7.

$$H(e^{j\Omega}) = \frac{r - e^{-j\Omega}}{1 - re^{-j\Omega}} = \frac{r - (\cos(\Omega) - j \sin(\Omega))}{1 - r(\cos(\Omega) - j \sin(\Omega))} = \frac{(r - \cos(\Omega)) + j \sin(\Omega)}{(1 - r \cos(\Omega)) + j r \sin(\Omega)}$$

$$|H(e^{j\Omega})|^2 = \frac{(r - \cos(\Omega))^2 + \sin(\Omega)^2}{(1 - r \cos(\Omega))^2 + r^2 \sin(\Omega)^2} = \frac{r^2 - 2r \cos(\Omega) + 1}{1 - 2r \cos(\Omega) + r^2 (\cos^2(\Omega) + \sin^2(\Omega))} = 1$$

5.8. See Example 5.4.

5.9. The coefficients a_0 , a_1 , a_2 , b_1 , & b_2 are clearly not integers, and if we just round each to the nearest integer or take its integer part, the result will be rather silly. So we must choose a scaling factor K which is a large integer, say 100 or 1000 or 1024 or maybe 100000. We multiply each coefficient by K and then round to the nearest integer. The effect of rounding is now less drastic, and we can compensate later for the scaling up of the coefficients by dividing by K . Clearly the larger K , the less serious will be the effect of rounding. However the integers produced must not be too large as overflow may occur. In a 16-bit microprocessor like the TMS32010, stored integers representing filter coefficients are limited to the range -32768 to 32767 . Similarly signal values from the ADC lie between -32768 to 32767 . The processor can multiply together two 16-bit words (e.g. a signal sample and an integerised filter coefficient) to produce a 32-bit result, and can add 32-bit numbers together. But any resulting 32-bit number must be scaled back to 16-bits before it can be stored as a signal and subjected to further multiplication processes, or output to a DAC. The scaling back to 16 bits is achieved by dividing by K . It is also much easier to divide by a constant K which is a power of two, such as 1024 than to divide by say 100 or 1000. Since second order IIR section filter coefficients normally lie between -2 and $+2$, choosing $K=1024$ is fairly safe, though not necessarily optimal. Having chosen K , we then calculate the integerised coefficients as follows:

$$IB1 = \text{int}(K*b1); IB2 = \text{int}(K*b2); IA0 = \text{int}(K*a0); IA1 = \text{int}(K*a1); IA2 = \text{int}(K*a2);$$

Now we can write the program:-

```

IW1:=0; IW2:=0;           (these are 16-bit integers)
L: Input IX;              (16-bit integer from ADC)
  P := IX*K - IB1*IW1 - IB2*IW2; (32-bit result in P)
  IW := P / K;            (integer divide by shifting to produce the 16-bit IW)
  P := IW*IA0 + IW1*IA1 + IW2*IA2; (32-bit result in P)
  IY := P / K;           (integer divide by shifting to produce the 16-bit IY)
  Output IY;             (send 16-bit result to DAC)
  IW2 := IW1;
  IW1 := IW;
  Goto L                 (Go back for next sample)

```

These program steps are easily understood and converted to a different language such as C or assembly language. Apologies for the “goto” statement.

The program given above may be understood in a more professional way by defining $IB1$, $IB2$, etc. to be “Q-12” fixed-point numbers; i.e. a decimal point (strictly a “binary” point) would be assumed to exist after the most significant 4 bits. If the programmer chooses to define IX as a Q-12 number also, P becomes a Q-24 number which is scaled back to a Q-12 number by the statement $IY=P/K$. The programmer must remember the Q-formats assumed for each word and keep track of what happens at each stage of the calculation. Thinking about Q-factors rather than scaling by K constants is ultimately more elegant and flexible, but students tend to prefer K constants to begin with. You can generate the same code by either mode of thinking. Clearly good documentation is going to be very important as calculations get complicated. Fixed point programming is important for mobile communications since it simplifies the hardware and computational complexity (though not the programming effort) and this leads to power savings and longer battery life.

5.10. If the order is as in the question, the impulse response of the combination is $\{h_1[n]\} \otimes \{h_2[n]\}$.

If the order is reversed, the impulse response becomes: $\{h_2[n]\} \otimes \{h_1[n]\}$. This is equal to $\{h_1[n]\} \otimes \{h_2[n]\}$ as we know.

- 5.11. Notch frequency is $\pi/2$. Place zero on unit circle at $z = \exp(j\pi/2)$ and its complex conjugate at $z = \exp(-j\pi/2)$. Place poles at $z = (1-\alpha) \exp(j\pi/2)$ and $z = (1-\alpha) \exp(-j\pi/2)$. If α is small, the 3dB bandwidth is 2α . Therefore $2\alpha = 3.2 * 2\pi/200$ and $\alpha = 0.05$.

$$H(z) = \frac{(z - \exp(j\pi/2))(z - \exp(-j\pi/2))}{(z - 0.95\exp(j\pi/2))(z - 0.95\exp(-j\pi/2))} = \frac{z^2 + 1}{z^2 + 0.95^2} = \frac{1 + z^{-2}}{1 + 0.9025 z^{-2}}$$

Hence etc.

Section 6

- 6.1. Refer to general formula.

- 6.2. Multiply out the denominator in 6.1, then to scale the cut-off frequency from 1 radian/second to ω_c , replace s by s/ω_c .

$$y(t) - (2/\omega_c) dy(t)/dt - (2/\omega_c^2) d^2y(t)/dt^2 - (1/\omega_c^3) d^3y(t)/dt^3 = x(t)$$

$\omega_c = 1000\pi$. Sampling interval $T = 0.0001$ seconds. $T\omega_c = \pi/10$.

$$H(z) = \frac{1}{1 + 20(1-z^{-1})/\pi + 200(1-2z^{-1}+z^{-2})/\pi^2 + 1000(1-3z^{-1}+3z^{-2}-z^{-3})/\pi^3}$$

$$= \frac{1}{1 + 6.366(1-z^{-1}) + 20.264(1-2z^{-1}+z^{-2}) + 32.2515(1-3z^{-1}+3z^{-2}-z^{-3})}$$

$$H(z) = \frac{1}{59.8815 - 143.649z^{-1} + 117.019z^{-2} - 32.2515 z^{-3}}$$

I have not checked this yet.

- 6.3.

$$\Omega_c = \pi/2 \text{ radians/sample.}$$

$$\text{Pre-warped analogue frequency: } 2 \tan(\Omega_c/2) = 2 \text{ radians/second.}$$

Required analogue prototype transfer function is :

$$H(s) = \frac{1}{(1 + s/2)(1 + s/2 + s^2/4)}$$

Replacing s by $2(z-1)/(z+1)$ and rearranging, we obtain:

$$H(z) = (1/6)(1 + z^{-1}) \frac{(1 + 2z^{-1} + z^{-2})}{(1 + (1/3)z^{-2})}$$

- 6.5. Cut-off frequency is $\Omega_c = \pi/2$.

Must be -20 dB or less at $\Omega = 3\pi/4$.

Analogue prototype transfer function must have cut-off $\omega_c = 2\tan(\pi/4) = 2$ radians/second.

Gain of analogue prototype must be -20 dB or less at $\omega = 2\tan(3\pi/8) = 4.828$ radians/second.

Gain of an n^{th} order Butterworth low-pass analogue filter with cut-off frequency $\omega_c = 2$ radians/second is:

$$G(\omega) = \frac{1}{\sqrt{(1 + (\omega/2)^{2n})}}$$

So we need to find the smallest integer value of n such that $20 \log_{10}(G(4.828)) < -20$ dB.

This means that we must have $\log_{10}((1 + (2.414)^{2n}) > 2$

$$\text{i.e. } (1 + (5.827)^n) > 10^2 = 100$$

$$5.827^n > 99 \quad \text{Smallest possible integer value of } n \text{ is three.}$$

We can now design the filter by applying the bilinear transformation to 3^{rd} order $H(s)$ with cut-off $\omega_c = 2$.

Section 7

7.1. See recommended textbook.

7.2. Yes it is possible. We can "over-sample" to simplify the analogue filters required to avoid aliasing. Then we can digitally filter the resulting over-sampled signal so that we can then reduce the sampling rate. Lowering the sampling rate allows more efficient processing, storage and transmission. The digital low-pass filter operating on the over-sampled signal can be much sharper and more reliable (no variations with temperature, manufacturing tolerance or aging, for example) than an equivalent analogue filter.

7.3. "Images" (see lecture notes) are increased in frequency. They are further away in frequency from the signal itself, and are hence easier to filter out without significantly affecting the signal.

7.4. QN power in range -10 kHz to 10 kHz: $\Delta^2/12$.

$$\text{Max sine-wave amplitude: } 2^8 \Delta^2 / 2 \quad \text{Max power: } (2^7 \Delta^2) / 2$$

$$\text{Max SQNR: } 2^{13} \Delta^2 / (\Delta^2 / 12)$$

$$= 12 \times 2^{13} \text{ in range } -10\text{kHz to } 10\text{kHz.}$$

Signal range is -4 kHz to 4 kHz, so can filter off QN above 4 kHz.

$$\text{SQNR in range } -4\text{kHz to } 4\text{kHz} : 12 \times 2^{13} / 0.4 ; \text{ i.e. } 53.9 \text{ dB.}$$

Replacing 8 bit by 10-bit ADC adds 12 dB. Decreasing sampling rate to 10 kHz means that we can no longer divide by 0.4 . Answer is 61 dB., Need more expensive ADC and sharper analogue anti-aliasing filter.

7.5. By argument in notes (p.2.6) $|H(j\omega)| = |\text{sinc}(\omega T / (8\pi))| \times 0.25$.

$$\text{When } \omega = \pi/T \text{ (=half sampling freq) } |H(j\omega)| = |\text{sinc}(1/8)| \times 0.25 = 0.24$$

$$\text{When } \omega = 0, |H(j\omega)| = 0.25$$

Therefore gain only falls from 0.25 to 0.24 , ie from -12 dB to -12.4 dB

We only drop by 0.4 dB as ω goes from zero to half the sampling frequency.

Disadvantage: reduced signal-to-noise ratio. This can be corrected by increasing pulse height but this means that we have very high voltages.

Bookwork . See notes.

7.6. See notes for explanation for zero order hold of "sample & hold" reconstruction.

$$H(j\omega) = e^{-j\omega T/2} \text{sinc}(\omega T/(2\pi))$$

7.7. In notes

7.8. Ideally we need a digital filter with $|H(e^{j\Omega})| = 1 / \text{sinc}(\Omega/2)$

Can we design an FIR filter by windowing method? The inverse DTFT gives us a formula that is probably too hard to integrate. Various alternative approaches exist such as sampling $1/\text{sinc}(\Omega/2)$ in the frequency domain and using the DFT or FFT in place of the inverse DTFT to perform the inverse transform.

Alternatively we can approximate $|H(e^{j\Omega})| = 1 / \text{sinc}(\Omega/2)$ by a simpler function such as the linear function:

$$|H(e^{j\Omega})| = 1 + \Omega/(2\pi)$$

7.9. Must have $20 \log_{10}(1/\sqrt{1 + (\omega_a / \omega_c)^2}) \leq -37$ dB. Therefore $\omega_a \geq 889.5 \times 10^3$ radians/second
If aliasing is going to affect our 0-2kHz useful signal, its frequency must be greater than 141.6 kHz, otherwise it would not be attenuated sufficiently by the filter.

If the sampling rate is f_s any noise above $f_s/2$ will be aliased, but it will not affect the signal unless the result lies between 0 and 2kHz. The lowest frequency that will cause problems is $f_s - 2$ kHz.

Therefore we must ensure that $f_s - 2$ kHz \geq 141.6 kHz.

Therefore, minimum sampling freq is 143.6kHz for original question.

Section 8

For problems on Section 8 and their solutions, refer to past examination papers.