CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year: 2007/2008

Examination Period: Autumn

Examination Paper Number: CM0340SOLNS

Examination Paper Title: Multimedia
SOLUTIONS

Duration: 2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are 13 pages.
There are 4 questions in total.
There are no appendices.
The maximum mark for the examination paper is 80 and the mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:
ONE answer book.

Instructions to Students:

Answer 3 questions.
The use of translation dictionaries between English or Welsh and a foreign language bearing an appropriate departmental stamp is permitted in this examination.

Q1. (a) *Why is* data compression *desirable for multimedia activities?*

Data is high resolution in both temporal and/or spatial domains. [1]
Storage, bandwidth and processing limitations need compressed data. [1]

2 Marks total — BOOKWORK

(b) *What is the distinction between lossy and lossless data compression?*

Lossless Compression – Where data is compressed and can be reconstituted (uncompressed) without loss of detail or information. These are referred to as bit-preserving or reversible compression systems also. [1]

Lossy Compression – Where the aim is to obtain the best possible fidelity for a given bit-rate or minimizing the bit-rate to achieve a given fidelity measure. Video and audio compression techniques are most suited to this form of compression. [1]

2 Marks total — BOOKWORK

(c) *What are the main* differences *between the target media for JPEG and GIF compression?*

JPEG — 24 bit full/true colour imagery, typically photographic in nature. [1]
GIF — 8-bit colour imagery, suits graphics type images better. [1]

2 Marks total — BOOKWORK

(d) *What improvement did the LZW algorithm make on previous LZ versions?*

LZW introduced the idea that only the initial dictionary needs to be transmitted to enable **decoding**. [1]
The decoder is able to build the rest of the table from the encoded sequence. [1]

2 Marks total — BOOKWORK

(e) *Describe the* LZW *algorithm for encoding an input sequence, giving suitable pseudocode.*

The LZW Compression Algorithm can summarised as follows:

```
w = NIL;
while ( read a character k )
  {
         if wk exists in the dictionary
             w = wk;
         else
             { add wk to the dictionary;
                output the code for w;
                 w = k;
              }
   }
```

[6]

May have a prior dictionary

- Original LZW used dictionary with 4K entries, first 256 (0-255) are ASCII codes.
- GIF builds dictionary of‘ “image blocks” no priors.

[1]

6 Marks for algorithm
1 Mark for dictionary build

7 Marks total — BOOKWORK

(f) *Given an initial dictionary:*

Index	*Entry*
1	a
2	b
3	h
4	i
5	s
6	t

and output of an LZW encoder*:*

6 3 4 5 1 3 1 6 2 9 11 16

decode *the above sequence (which is not intended to represent meaningful English*

The LZW Decompression Algorithm is as follows:

```
read a character k;
output k;
w = k;
while ( read a character k )
/* k could be a character or a code. */
  {
           entry = dictionary entry for k;
           output entry;
           add w + entry[0] to dictionary;
           w = entry;
  }
```

Process as follows:

- Read 6 from input, lookup 6 so output t. Set $w = t$
- Read 3 from input, lookup 3 so output h. Add entry 7 to dictionary th. Set $w = h$
- Read 4 from input, lookup 4 so output i. Add entry 8 to dictionary hi. Set $w = i$
- Read 5 from input, lookup 5 so output s. Add entry 8 to dictionary is. Set $w = s$
- Read 1 from input, lookup 1 so output a. Add entry 9 to dictionary sa. Set $w = a$
- Read 3 from input, lookup 3 so output h. Add entry 10 to dictionary ah. Set $w = h$
- Read 1 from input, lookup 1 so output a. Add entry 11 to dictionary ha. Set $w = a$
- Read 6 from input, lookup 6 so output t. Add entry 12 to dictionary at. Set $w = t$
- Read 2 from input, lookup 2 so output b. Add entry 13 to dictionary tb. Set $w = b$
- Read 9 from input, lookup 9 so output sa. Add entry 14 to dictionary bs. Set $w = sa$

- Read 11 from input, lookup 11 so output ha. Add entry 15 to dictionary sah. Set $w = ha$
- Read 16 from input, lookup 16 so output $ha....$ — We don't know full entry yet but we know it starts with current $w = ha....$ Add entry 16 to dictionary hah. **We now have entry 16 —** this is the built in '*exception handler*' of LZW, so we can output 16 now. Set $w = hah$

Final Decode Table:

Index	Entry
1	a
2	b
3	h
4	i
5	s
6	t
7	th
8	hi
9	sa
10	ah
11	ha
12	at
13	tb
14	bs
15	sah
16	hah

Output decode sequence:

$$thisahatbsahahah$$

6 Marks for correct algorithm and applying it.
4 marks for showing table constructions and the correct decoded sequence
2 Marks for solving the LZW '*exception handler*' look ahead last step.

12 Marks total — UNSEEN

27 Marks Total Question 1

Q2. (a) *In a digital signal processing system, what is meant by* block *and* sample-by-sample *processing.*
Give **one** *example of an application of* **each type**.

Block processing : data is transferred into a **memory buffer** and then processed each time the buffer is filled with new data. [1]
E.g. fast Fourier transforms (FFT), Discrete Cosine Transform (DCT), convolution, convolution reverb — **more soon** [1]

Sample-by-sample processing : input is processed on individual sample data. [1]
E.g. volume control, envelope shaping, modulation (ring/amplitude), IIR/-FIR filtering.... [1]

4 Marks Total — Bookwork

(b) *In a digital signal processing system, what is meant by a* linear *and a* non-linear *time invariant system.*
Give **one** *example of an application of* **each type**.

Linear time invariant system (LTI) : Systems that **do not change** behaviour over time and satisfy the superposition theory. The output signal is signal changed in amplitude and phase. [1]
E.g. Convolution, Filters [1]

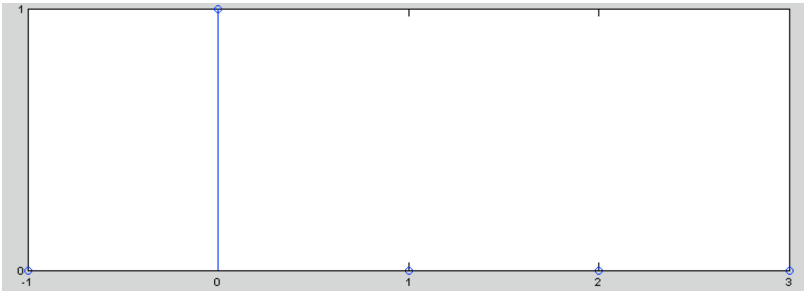
Non-linear time invariant system : Systems whose output is strongly shaped by non-linear processing that introduces harmonic distortion — *i.e.* harmonics that are not present in the original signal will be contained in the output. [1]
E.g. Limiters, Compressors, Exciters, Distortion, Enhancers. [1]

4 Marks Total — Bookwork

(c) *Give the definition of an* impulse response.
Give **two** *practical uses of an impulse response in digital signal processing.*
E.g. Unit Impulse:

- Defined as:

$$\delta(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise } (n \neq 0) \end{cases}$$



[2]

Two Uses:

- A test signal for digital systems, characterise system by their response functions. [1]
- Convolution Reverb — sample a rooms impulse response, convolve with input to get a reverberated output. [1]

4 Marks Total — Bookwork

(d) *List the three basic components used in constructing a* signal flow graph.

3 Components:

- Delay [1]
- Multiplication [1]
- Summation [1]

3 Marks Total — Bookwork

Why is it desirable to describe systems using these components.

- Componentisation of nearly all basic DSP filters, delays into standard processes. [1]
- Basic building blocks of nearly all basic DSP filters, delays description of algorithms simple. [1]
- These component make easy cost-effective construction of hardware implementations from standard components. [1]

3 Marks Total — Unseen/Assimilation of bookwork/discussed in tutorial
6 Marks Total For Question (d) Subpart

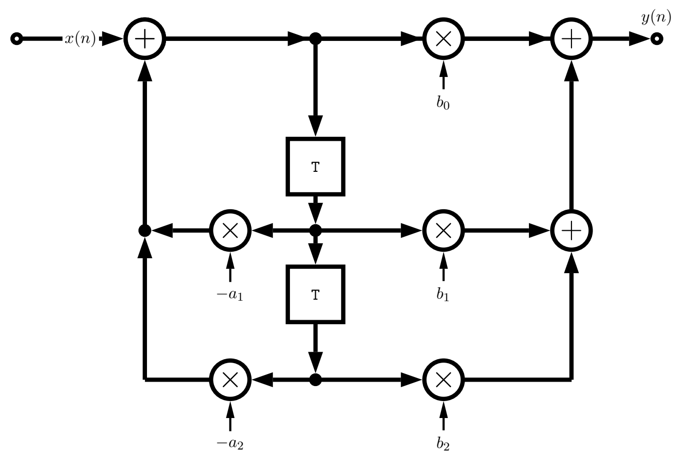
(e) *What is the* **main** *distinction between an* infinite impulse response (IIR) *and a* finite impulse response (FIR) *filter.*

IIR filters have a **feedback loop** in their construction

1 Mark Bookwork

(f) Given the following difference equation construct its *signal flow diagram*:

$$\begin{aligned} y(n) &= b_0x(n) + b_1x(n-1) + b_2x(n-2) \\ &\quad -a_1y(n-1) - a_2y(n-2) \end{aligned}$$



[8]

8 Marks — Unseen.
Possible other less efficient solutions with more than two delay units (not shared for $y(n-1)/x(n-1)\dots$ tapping award less marks if so

27 Marks Total Question 2

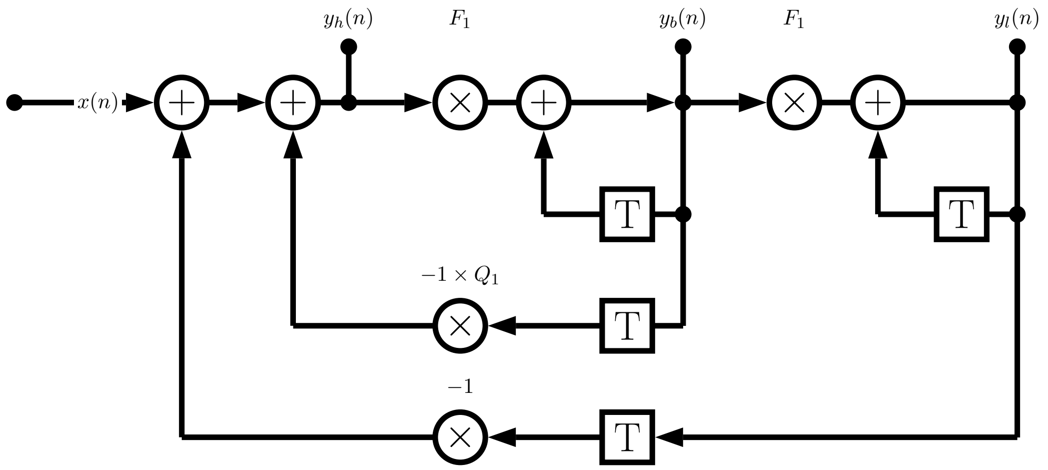
Q3. (a) *List* **six** *broad classes of* digital audio effect. *Give an example effect of* **each** *type of effect.*

6 broad classes:

Basic Filtering — Lowpass, Highpass filter etc,, Equaliser [1]
Time Varying Filters — Wah-wah, Phaser [1]
Delays — Vibrato, Flanger, Chorus, Echo [1]
Modulators — Ring modulation, Tremolo, Vibrato [1]
Non-linear Processing — Compression, Limiters, Distortion, Exciters/Enhancers [1]
Spacial Effects — Panning, Reverb, Surround Sound [1]

6 Marks — Bookwork

(b) *Give a description, including a signal flow diagram and algorithm, of the* state variable filter.



where:

$$
\begin{aligned}
x(n) &= \text{input signal} \\
y_l(n) &= \text{lowpass signal} \\
y_b(n) &= \text{bandpass signal} \\
y_h(n) &= \text{highpass signal}
\end{aligned}
$$

[4]

The state variable filter algorithm/difference equations are given by:

$$
\begin{aligned}
y_l(n) &= F_1 y_b(n) + y_l(n-1) \\
y_b(n) &= F_1 y_h(n) + y_b(n-1) \\
y_h(n) &= x(n) - y_l(n-1) - Q_1 y_b(n-1)
\end{aligned}
$$

[2]

with tuning coefficients F_1 and Q_1 related to the cut-off frequency, f_c, and damping, d:

$$F_1 = 2\sin(\pi f_c/f_s), \qquad \text{and } Q_1 = 2d$$

[2]

8 Marks — Bookwork

(c) *Give* **two** *advantages of the* state variable filter.

The 2 advantages are:

i. Independent control over the cut-off frequency and damping factor of a filter. [1]
ii. Simultaneous lowpass, bandpass and highpass filter output. [1]

2 Marks — Bookwork

(d) *A* band-reject *filter is a filter which passes all frequencies except those in a stop band centered on a center frequency. How can such a filter be implemented using* **two** *state variable filters?*

- Create a low pass filter with cut-off frequency, f_{low} [1]
- Create a high pass filter with cut-off frequency, f_{high} [1]
- Add the outputs of the low and high pass filter. [1]
- Band reject is the range f_{low}—f_{high}, where $f_{low} < f_{high}$ [1]

4 Marks — UNSEEN.
Band-reject/bandstop filter defined in lecture but not implemented

(e) *How may a* phaser *effect be implemented using* **two** *state variable filters?*

- A **phaser** is implemented with a (set of) *time-varying* frequency *notch* filters. [2]
- *Notch* filter a very narrow *band-reject/bandstop* filter. [1]
- To get narrow band,:
  - set high Q factor — sharp slope on filter cut-off [1]
  - $f_{low} < f_{high}$, but f_{low} close in value to f_{high} (around some centre frequency) [1]
- Perform band-reject with above parameters, as in part (d) above **BUT** modulate the frequency range with a sine or triangular wave over a short range. [1]

- A *cascade* of such filters implements an multiple notch phases where each notch filter has a different centre frequency. [1]

7 Marks — UNSEEN.
Phaser defined in lecture but no implementation given — similar to wah-wah (notch filter instead of bandpass) implementation which was given in lecture. Notch filter defined in lecture as very narrow band-reject/bandstop filter

27 Marks Total Question 3

Q4. (a) *Give a definition of a* one-dimensional Fourier transform.

The Fourier transform of that function is denoted $F(u)$, where u represents spatial frequency is defined by

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x u} \, dx.$$

[2]

2 Marks - BOOKWORK

(b) *Explain in detail how data is represented after the Fourier transform has been applied to a signal.*

Essentially the Fourier transform decomposes an input signal into a series of sine waveforms with varying amplitude, frequency and phase. [2]

2 Marks - BOOKWORK

(c) Outline the basic approach to performing data *filtering* with the Fourier transform.

- Compute Fourier transform of signal. [1]
- Compute Fourier transform of Filter (if convolution) or create filter in frequency space (e.g. ideal low pass/butterworth filter). [1]
- Multiply (or divide if forward filter form used above) the two Fourier transforms above. [1]
- Inverse Fourier transform to get filtered data. [1]

4 Marks - BOOKWORK

(d) Describe **one** application of Fourier transform filtering methods in multimedia data compression.

Most obvious is MPEG AUDIO, (however one alternative answer might be in JPEG/MPEG video if Fourier Transform is replaced by a Discrete Cosine Transform which is a related method?)

MPEG audio compression basically works by:

- Dividing the audio signal up into a set of 32 frequency subbands — apply FOURIER FILTERING [2]
- Subbands approximate **critical bands of human hearing**. [1]
- Each band quantised according to the '*audibility*' of *quantisation noise*. [1]

- Exploit Frequency Masking — near frequencies not heard in same time frame [2]
- Exploit Temporal Masking — near frequencies not heard close to some short time frame between frequencies [2]

8 Marks - (Distillation of) BOOKWORK

(e) *An* exciter *is a digital audio signal process that emphasises or de-emphasises certain frequencies in a signal in order to change its* timbre*. Describe how you could use the Fourier transform to implement such a process, giving a* practical example *and explaining how it works.*

Approach similar to basic filtering except no frequency content removed. [2]
In some cases similar to an equaliser (discussed in lectures) — but non-linear exaggeration of frequencies. [1]

Basic Approach:

- Compute Fourier transform of signal. [1]
- Apply some function that will enhance/diminish certain frequencies in frequency/Fourier space. [6]
  Example to enhance high frequencies: This could be similar to distortion type amplification (discussed in lectures) which operates on amplitude not frequencies.
  E.g:
  Some *soft clipping*

$$f(x) = \begin{cases} x & \text{for} \quad 0 \le x < 1/3 freqrange \\ \frac{3-(2-3x)^2}{3} & \text{for} \quad 1/3 \le x < 2/3 freqrange \\ 1 & \text{for} \quad 2/3 \le x \le 1 freqrange \end{cases}$$

  or some non-linear function:

$$f(x) = \frac{x}{|x|}(1 - e^{\alpha x^2/|x|})$$

- Inverse Fourier transform to get excited data. [1]

11 Marks - UNSEEN. *exciter* mentioned briefly in lecture but no implementation details discussed

27 Marks Total Question 4

END OF EXAMINATION