Speech Processing 2001

EE & MSc

Answer FOUR questions in 3 hours.

There are SIX questions on this paper.

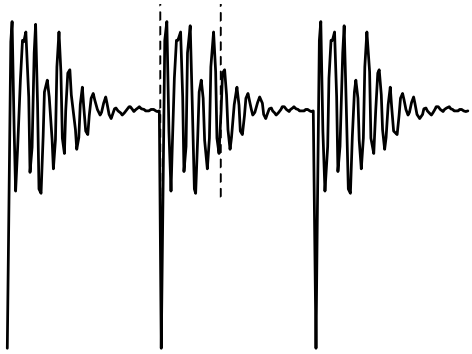


Figure 1

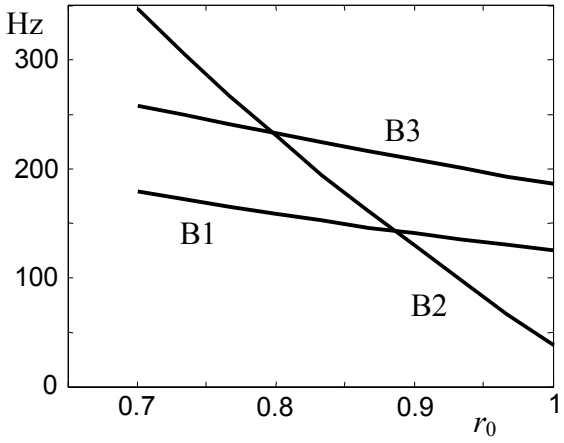


Figure 2

Sample	30	33	36	39	42	45	48	51	54	57	60	63
Value	-187	167	-167	94	-103	66	-23	52	-37	8	-21	14

Table 1

[These figures and table relate to question 1]

1. *Figure 3* shows a lossless tube model of the vocal tract with the glottis at the left and the lips at the right. U_G, U_1 and U_L are the *z*-transforms of the forward acoustic waves at the glottis, the glottis end of the first tube segment and the lips respectively. V_G, V_1 and V_L are the z-transforms of the corresponding reverse waves. A_G and A_1 are respectively the effective cross-sectional areas of the glottis and of the first tube segment.

   (a) Given the volume and pressure continuity relations:

   $$U_G - V_G = U_1 - V_1 \qquad \text{and} \qquad \frac{U_G + V_G}{A_G} = \frac{U_1 + V_1}{A_1}$$

   show that U_G may be expressed in matrix form as

   $$U_G = \frac{1}{1 + r_0}\begin{pmatrix} 1 & -r_0 \end{pmatrix}\begin{pmatrix} U_1 \\ V_1 \end{pmatrix}$$

   and derive an expression for the reflection coefficient, r_0, in terms of A_G and A_1. [7]

   (b) Explain what value you would expect for r_0 during the closed-glottis interval and how you would expect it to vary during voiced speech as the glottis opens and closes. Explain why it is sometimes desirable to restrict LPC analysis to the closed-glottis intervals of the speech signal. [4]

   (c) *Figure 1* shows the waveform of a vowel whose first three formants are at 420, 1333 and 1850 Hz respectively. The waveform is sampled at 8 kHz and the dashed lines indicate the closed-glottis interval (sample numbers 0 to 29) of the central larynx cycle. *Figure 2* shows how the bandwidths of the three formants (labelled B1, B2 and B3 respectively) vary with the glottal reflection coefficient, r_0.

   Table 1 gives the sample numbers and values of successive waveform maxima and minima taken from the open-glottis interval of this cycle. Estimate the value of r_0 during the open-glottis interval. You may assume without proof that the amplitude of a formant with bandwidth *b* Hz decays with a time constant of $(\pi b)^{-1}$ seconds. [9]

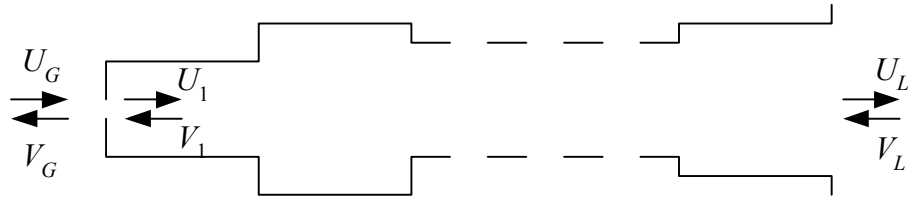


Figure 3

2. (a) State what is meant by a diphone-based concatenative text-to-speech synthesiser. Explain why a necessary part of such a synthesiser is a procedure for altering the pitch and duration of speech segments without affecting their formant frequencies. [4]

(b) A speech signal consists of a steady vowel sound lasting for 1 second with a sample rate of 8 kHz. The 8000 samples are numbered 0 to 7999. The pitch of the vowel is a constant 100 Hz with pitch marks (i.e. the energy maximum within each pitch cycle) located at sample numbers 40, 120, 200, …, 7960.

Explain the effect on the pitch, formant frequencies and duration of the signal when each of the following transformations is applied (in each case, the transformation is applied to the original signal):

(i) Inserting a duplicate of every fourth pitch cycle, i.e. outputting samples in the following order: 0:319, 240:639, 560:959, … .

(ii) Removing the central 10 samples of each pitch cycle, i.e. outputting samples, 5:74 ,85:154, 165:234, … . [8]

(iii) Outputting the samples at a sample frequency of 10 kHz instead of 8 kHz.

(c) It is desired to change the pitch to 176 Hz, increase all formant frequencies by 10% and change the signal duration to 0.852 seconds. Describe the sequence of operations required and state the output sample frequency required. Assuming that the first output pitch mark is at sample number 40, determine which two input samples contribute to the output sample number 150. [8]

3. (a) A speech signal is represented by the samples $s(n)$, n = 0, 1 ,…, N–1. The prediction error of a p^{th} order linear predictor is defined by $E = \sum_{n=p}^{N-1}\left(s(n) - \sum_{k=1}^{p} a_k s(n-k) \right)^2$ where the a_k are the prediction coefficients.

Show that the prediction coefficients that minimize E satisfy $\mathbf{Ra} = \mathbf{b}$ where the $(i,j)^{\text{th}}$ element of $\mathbf{R}$ is given by $r_{i,j} = \sum_{n=p}^{N-1} s(n-i)s(n-j)$, the i^{th} element of $\mathbf{b}$ is given by $b_i = r_{i,0}$ and $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}^T$. [6]

(b) Explain the difference between covariance and autocorrelation LPC and state their relative advantages. Explain why most speech coders use autocorrelation LPC. [6]

(c) A signal sampled at 8 kHz consists of white noise plus three cosine waves at 800 Hz, 1 kHz and 1.2 kHz of relative amplitudes 1, 1 and 0.2. *Figure 4* shows the filter spectra, with a normalised frequency axis, resulting from 10th order LPC analysis under the following conditions (not necessarily in this order):

(i) Covariance LPC with frame lengths of 3.5 ms and 80 ms

(ii) Autocorrelation LPC using Hamming windows of length 3.5 ms and 80 ms.

Identify which plot corresponds with each of the four conditions. Give reasons for your choice and explain the factors that cause the differences between the plots. [8]

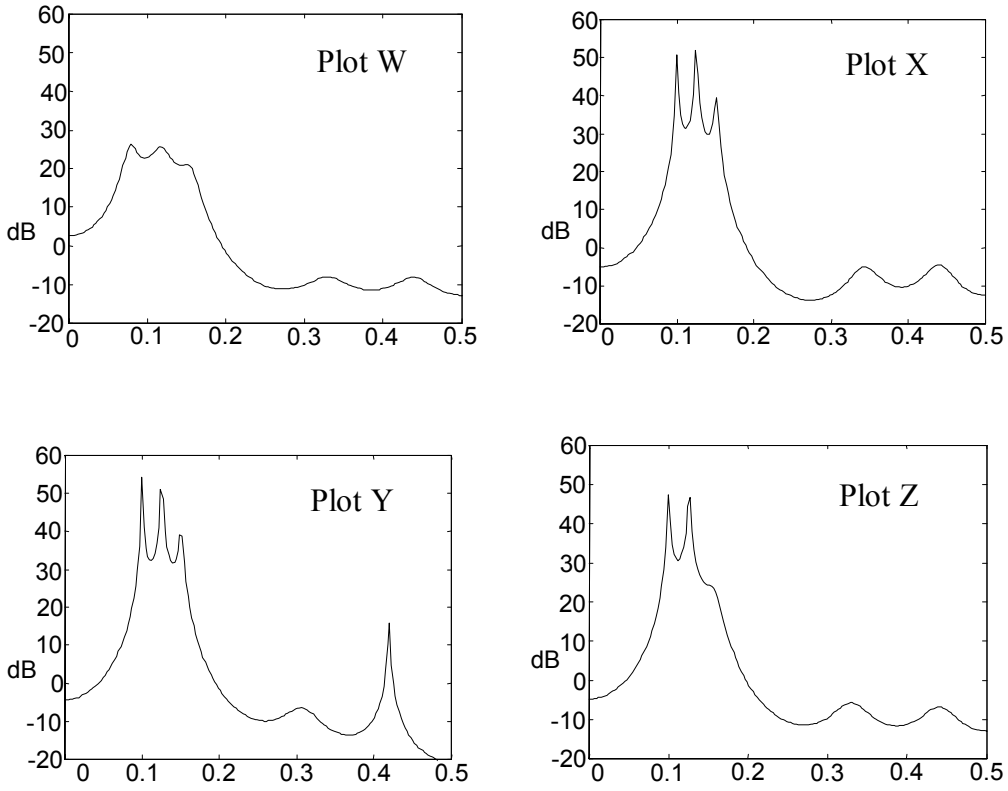


Figure 4 [All frequency axes are in normalized Hz]

4. (a) Outline the steps involved in the encoding of speech in a Code-Excited Linear Prediction (CELP) speech encoder. [5]

(b) *Figure 5* shows part of a CELP encoder in which the codebook has K entries of length N samples. The k^{th} codevector, $x_k(n)$, is passed through a filter, $H(z)$, multiplied by a gain, g_k, and subtracted from a target $t(n)$ to generate the error signal $e_k(n)$. The filter output is defined by

$$y_k(n) = \sum_{i=0}^{n} x_k(i)h(n-i) \qquad \text{for } n = 0,\ldots, N-1$$

where $h(i)$ is the impulse response of $H(z)$.

Show that $E_k = \sum_{n=0}^{N-1} e_k^2(n)$ is minimized when $g_k = \dfrac{\sum_{n=0}^{N-1} y_k(n)t(n)}{\sum_{n=0}^{N-1} y_k^2(n)}$ and that, when g_k equals this value, $E_{k(opt)} = \sum_{n=0}^{N-1} t^2(n) - \dfrac{\left(\sum_{n=0}^{N-1} y_k(n)t(n)\right)^2}{\sum_{n=0}^{N-1} y_k^2(n)}$. [5]

(c) In order to reduce the computational complexity, the codebook elements are restricted as follows:

- all but two elements of $x_k(n)$ must equal zero.
- the two non-zero elements, $x_k(a_k)$ and $x_k(b_k)$ may only equal +1 or –1.

Show that E_k may now be expressed as

$$E_{k(opt)} = \sum_{n=0}^{N-1} t^2(n) - \frac{\left(x_k(a_k)s(a_k) + x_k(b_k)s(b_k)\right)^2}{c(a_k,a_k) + 2x_k(a_k)c(a_k,b_k)x_k(b_k) + c(b_k,b_k)}$$

where the vector $s(n)$ and the matrix $c(m,n)$ are independent of k. Give expressions for $s(n)$ and $c(m,n)$. [5]

(d) Estimate the number of multiplications required to calculate all the elements of $s(n)$ and $c(m,n)$. [5]

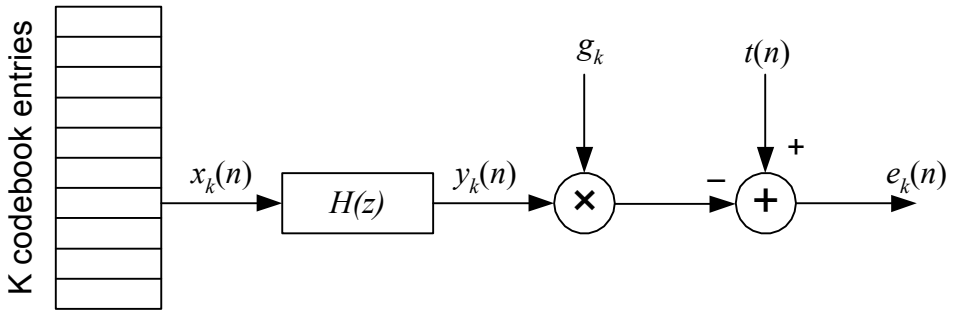


Figure 5

5. (a) In a statistical speech recogniser, $pr(w|s)$ denotes the conditional probability of a particular word sequence, w, given an input speech utterance, s. Describe how Bayes' Theorem may be used to express this probability in terms of a *language model* and an *acoustic model of speech production*. [2]

(b) Explain what is meant by a *unigram*, a *bigram* and a *trigram* language model in a speech recogniser. In each case give the number of transition probabilities that must be estimated for a recogniser with a vocabulary of 20,000 words. [3]

(c) In a security system, passwords consist only of the digits 1, 2 and 3 and may be of any length providing they contain at least one digit. Thus "113", "2", "333" and "1212321" are all valid passwords. *Table 2* gives the frequencies of each possible pair of successive words in a representative sample of passwords. "Start" and "end" represent the start and end of a password. Calculate the unigram probabilities for each of 1, 2 , 3 and "end". [4]

(d) Bigram probabilities are calculated according to the following formula:

$$p(i,j) = \begin{cases} \dfrac{N(i,j) - d(i)}{N(i)} & \text{for } N(i,j) > 2 \\ b(i)\,p(j) & \text{for } N(i,j) \le 2 \end{cases}$$

where $p(i,j)$ is the bigram probability of word j given word i, $p(j)$ is the unigram probability for word j and $N(i,j)$ and $N(j)$ are the corresponding occurrence counts in the training data. The discounting factor, $d(i)$ equals zero if $N(i,j) > 2$ for all valid next states j, and equals 0.5 if $N(i,j) \le 2$ for any valid j. The factor $b(i)$ is chosen so that

$$\sum_j p(i,j) = 1$$

Calculate the values of $d(i)$ and $b(i)$ for each i and all the bigram probabilities $p(i,j)$. [8]

(e) Explain why a different formula is used in (d) above according to whether $N(i,j)$ is above or below 2.

Explain why the formula for large $N(i,j)$ is $\dfrac{N(i,j) - d(i)}{N(i)}$ rather than $\dfrac{N(i,j)}{N(i)}$ and describe the effect this has on the bigram probabilities. [3]

$N(i,j)$		Second Word, j			
		1	2	3	end
Initial Word, i	start	10	10	30	0
	1	1	0	60	20
	2	3	50	2	10
	3	67	5	10	20

Table 2

6. (a) A speech utterance consists of T frames, $\mathbf{x}_1$, …, $\mathbf{x}_T$, and is compared with a Hidden Markov model having S states. The transition probability from state i to state j of the model is denoted by a_{ij} and the output probability density of frame t in state i is denoted by $d_i(\mathbf{x}_t)$.

$B(t,s)$ is defined to be the highest probability density that the model generates frames $\mathbf{x}_1$, …, $\mathbf{x}_t$ from any sequence of states for which frame 1 is in state 1 and frame t is in state s.

Explain fully how, for $t > 1$, $B(t,s)$ can be expressed in terms of $B(t–1,i)$ for i=1, 2, …, S. Indicate the values that should be given to $B(1,i)$. [4]

(b) Give a simplified expression for $\ln(d_i(\mathbf{x}))$ for the case when

$$d_i(\mathbf{x}) = (2\pi)^{-½P}|\mathbf{C}_i|^{-½}\exp\left(-½(\mathbf{x}-\mathbf{m}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}-\mathbf{m}_i)\right)$$

Explain why most recognition systems (i) perform their calculations with log probability densities rather than using the probability densities directly and (ii) assume that the matrix $\mathbf{C}$ is diagonal. [5]

(c) A 5-frame utterance is compared with a 4-state Hidden Markov model. *Table 3* shows the output log probability density, $\ln(d_i(\mathbf{x}_t))$, of each frame for each state of the model and *Figure 6* shows the state diagram of the model including the transition probabilities. Determine $\ln(B(5,4))$ and the state sequence to which it corresponds. [7]

(d) The model of *Figure 6* is used with output probability densities of the form given in part (b) to represent a certain word. Outline how the Viterbi training procedure can be used to estimate the model parameters from a number of training examples. [4]

	Input Frame				
	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$
State 1	-5	-9	-5	-5	-7
State 2	-4	-5	-7	-7	-6
State 3	-5	-3	-8	-8	-6
State 4	-6	-7	-8	-6	-8

Table 3

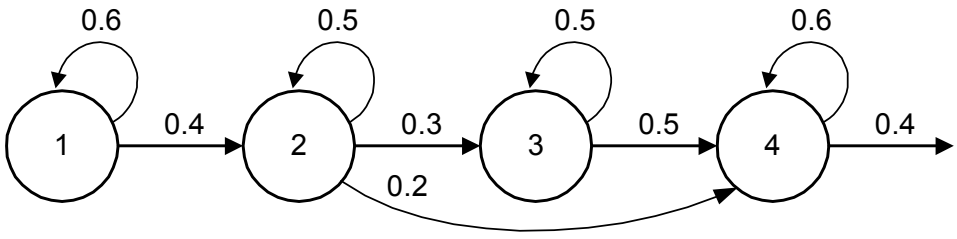


Figure 6

Solutions to Speech Processing 2001

1. (a) Substitue $V_G = U_G - U_1 + V_1$ into $A_1(U_G + V_G) = A_G(U_1 + V_1)$ to get

$$A_1(2U_G - U_1 + V_1) = A_G(U_1 + V_1)$$
$$2A_1U_G = U_1(A_G + A_1) + V_1(A_G - A_1)$$
$$U_G = \frac{1}{2A_1}\begin{pmatrix} A_G + A_1 & A_G - A_1 \end{pmatrix}\begin{pmatrix} U_1 \\ V_1 \end{pmatrix}$$

Hence $\frac{1}{1+r_0} = \frac{A_G + A_1}{2A_1} \quad \Rightarrow \quad r_0 = \frac{2A_1}{A_G + A_1} - 1 = \frac{A_1 - A_G}{A_1 + A_G}$

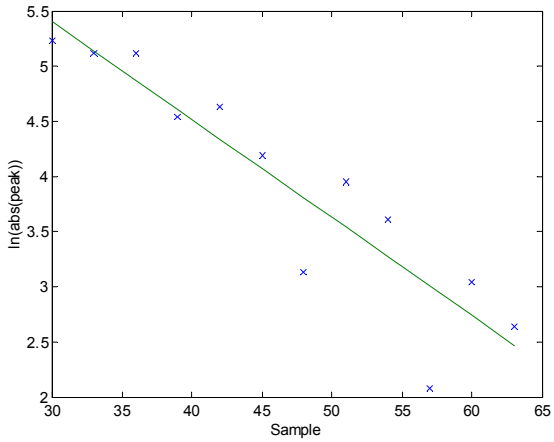
From which $U_G = \frac{1}{1+r_0}\begin{pmatrix} 1 & -r_0 \end{pmatrix}\begin{pmatrix} U_1 \\ V_1 \end{pmatrix}$ as required. [7]

(b) When $A_G = 0$ we have $r_0 = 1$ so this is the value we would expect during the closed phase. During the open-glottis interval, r_0 will decrease but will remain positive since A_G will never exceed A_1.

Restricting LPC to the closed phase allows a more accurate estimate of the vocal tract since the acoustic input is equal to zero. [4]

(c) From the values given in table 1, the period of the dominant oscillation is 6 samples; this corresponds to 1333 Hz and so arises from the second formant.

During the closed phase, we expect the envelope to decay exponentially. Therefore if we plot the absolute value of the peaks against time, we should get a straight line $y = k\exp(-\pi bt) = k\exp(-\pi bn / f_s) \quad \Rightarrow \quad \ln(y) = \ln(k) + n \times -\pi b / f_s$:



[9]

By estimating the gradient we get $-\pi b / f_s = -0.0888 \quad \Rightarrow \quad b = 226$ Hz

From the graph given in the question, we get $r_0 \approx 0.81$

2. (a) We create the output speech waveform by joining together segments of recorded speech. The base segments are diphones: i.e. they extend from the centre of one phoneme to the centre of the next.

In synthesised speech, the duration, amplitude and pitch of each phoneme must be controlled by the synthesiser. The pitch follows a smooth contour, generally rising to the first stressed syllable of a sentence and falling thereafter. The amplitude broadly follows the same contour as the pitch, rising slightly for each stressed syllable. The durations are adjusted so that the intervals between stressed syllables are roughly uniform. If the required values of duration, amplitude and pitch differ from those present in the recoded diphone, they must be adjusted accordingly. [4]

(b) The effects may be summarised as:

	Pitch	Formants	Duration
(i)	×1	×1	×1.25
(ii)	×1.14	×1	×0.875
(iii)	×1.25	×1.25	×0.8

[8]

(c) We can determine the required transformations by choosing the sample rate to get the right formants, reducing the cycle length to get the right pitch and finally replicating cycles to get the right duration:

	Pitch	Formants	Duration
Sample Freq ×1.1	×1.1	×1.1	×0.909
Remove 30/80 samples	×1.6	×1	×0.625
Repeat alternate cycles	×1	×1	×1.5
Total effect	×1.76	×1.1	×0.852

Thus the new output frequency is 8800 Hz

Since alternate cycles are replicated, the output pitch cycles centred at samples 40, 90, 140 and 190 are formed from the input cycles centred at samples 40, 120, 120 and 200. Output sample number 150 will be formed from the $+10^{th}$ sample of the input cycle centred at 120 and the -40^{th} sample of the input cycle centred at 200. It thus contains contributions from input samples 130 and 160. [8]

3. (a)

$$-\tfrac{1}{2}\frac{\partial E}{\partial a_i} = \sum_n \left(s(n) - \sum_{k=1}^{p} a_k s(n-k) \right) s(n-i)$$

$$= \sum_n s(n-i)s(n) - \sum_{k=1}^{p} a_k \sum_n s(n-i)s(n-k)$$

$$= r_{i,0} - \sum_{k=1}^{p} r_{i,k} a_k$$

[6]

Setting these partial derivatives to zero gives:

$$\sum_{k=1}^{p} r_{i,k} a_k = r_{i,0} \text{ for } i = 1, \ldots, p$$

or in matrix form **Ra** = **b**.

(b) Covariance LPC performs the summation of part (a) over a finite window [0,*N*–1]. No windowing of the signal is involved so there is no compromise between time and frequency resolution: we can get infinite frequency resolution with small data windows provided there is no noise. The window length, *N*, must be greater than the filter order, *p*, and in practice should be twice as long. The resultant filter is not guaranteed to be stable. Solving the equation for **a** requires order p^3 operations.

For autocorrelation LPC, the input signal is first multiplied by a window, e.g. a Hamming window, that tapers to zero (or near zero) at its ends. This imposes a tradeoff between time and frequency resolution: to obtain adequate frequency resolution for speech (100 Hz), the window length must be at least 20 ms. The summation of part (a) is then performed over $n = -\infty, \ldots, +\infty$ although all but a finite number of terms are zero. The resultant **R** matrix is toeplitz and the coefficients **a** can be found using the Levinson-Durbin algorithm with order p^2 operations. The filter will always be stable. [6]

(c) The 1200 Hz sinewave should have an amplitude $20\log_{10}(0.2) = -14$ dB.

Plot W has the poorest frequency resolution and broadest spectral peaks and is therefore autocorrelation LPC with a window length of 3.5 ms. This gives a frequency resolution of about 2/3.5 kHz = 571 Hz.

Plot Z is better, but the 1200 Hz sinewave has been almost overwhelmed by the sidelobes of the 1 kHz sinwave. This is therefore autocorrelation LPC with a window length of 80 ms.

Plot Y has accurately found the three sinewaves although their amplitudes are not quite right. It has also generated a spurious peak at around 3300 Hz. This is covariance LPC with a window length of 3.5 ms (28 samples) and the white noise has resulted in the false peak.

Finally Plot X is covariance LPC with a window length of 80 ms. The sinewaves have been accurately modelled and the remaining four poles have been used to create a noise floor at –10 dB. [8]

4. (a) The major steps are:

- LPC analysis of a frame. Usually autocorrelation LPC, usually bandwidth expanded, then converted to LSF coefficients then quantised or vector quantised prior to transmission. Often interpolated over several sub-frames.
- Define a perceptual weighting filter using the LPC spectrum. We want to down-weight noise near the formant frequencies since it is masked by the strong speech signal.
- Search for the optimum delay and gain of the long term predictor. Sometimes this uses fractional delays.
- Search for the optimum gain and stochastic codebook entry.
- Quantise the gains and transmit.

[5]

(b) We have: $e_k(n) = t(n) - g_k y_k(n)$

$$\tfrac{1}{2}\frac{\partial E_k}{\partial g_k} = \sum_n e_k(n)\frac{\partial e_k}{\partial g_k} = -\sum_n e_k(n)y_k(n) = g_k\sum_n y_k^2(n) - \sum_n y_k(n)t(n)$$

Setting this to zero gives $g_k = \dfrac{\sum_n y_k(n)t(n)}{\sum_n y_k^2(n)}$

From which:

$$E_{k(opt)} = \sum_n \big(t(n) - g_k y_k(n)\big)^2 = \sum_n t^2(n) - 2g_k\sum_n t(n)y_k(n) + g_k^2\sum_n y_k^2(n)$$

$$= \sum_n t^2(n) - 2\frac{\left(\sum_n t(n)y_k(n)\right)^2}{\sum_n y_k^2(n)} + \frac{\left(\sum_n t(n)y_k(n)\right)^2}{\sum_n y_k^2(n)} = \sum_n t^2(n) - \frac{\left(\sum_n t(n)y_k(n)\right)^2}{\sum_n y_k^2(n)}$$

[5]

(c) Since only two elements of $x_k(n)$ are non-zero, we have

$$y_k(n) = x_k(a_k)h(n-a_k) + x_k(b_k)h(n-b_k) \quad \text{where} \quad h(m) = 0 \text{ for } m < 0$$

Hence,

$$\sum_n t(n)y_k(n) = x_k(a_k)\sum_n t(n)h(n-a_k) + x_k(b_k)\sum_n t(n)h(n-b_k)$$

$$= x_k(a_k)s(a_k) + x_k(b_k)s(b_k) \quad \text{where} \quad s(m) = \sum_{n=m}^{N-1} t(n)h(n-m)$$

Similarly,

$$\sum_n y_k^2(n) = \sum_n \big(x_k^2(a_k)h^2(n-a_k) + 2x_k(a_k)x_k(b_k)h(n-a_k)h(n-b_k) + x_k^2(b_k)h^2(n-b_k)\big)$$

$$= \sum_n h^2(n-a_k) + 2x_k(a_k)x_k(b_k)\sum_n h(n-a_k)h(n-b_k) + \sum_n h^2(n-b_k)$$

$$= c(a_k,a_k) + 2x_k(a_k)x_k(b_k)c(a_kb_k) + c(b_k,b_k)$$

where $c(i,j) = \sum_{n=\max(i,j)}^{N-1} h(n-i)h(n-j)$

[5]

(d) (i) To calculate the element $s(m)$ requires $N–m$ multiplications, so calculating the entire vector requires

$$\sum_{m=0}^{N-1}(N-m) = ½N(N+1) \text{ multiplications}$$

(ii) Since $c(i,j)$ is symmetrical, we need only calculate it for i>=j.

We can calculate $c(i,j)$ recursively since

$$c(i,j) = c(i+1,j+1) + h(N-1-i)h(N-1-j)$$

[5]

Thus we initially calculate

$$c(N-1,j) = h(0)h(N-1-j) \quad \text{for} \quad j = 0,\cdots,N-1$$

and then calculate all the other $c(i,j)$ from the above recursion.

Each new element of $c(i,j)$ requires only one multiplication for a total of $½N(N+1)$.

5. (a) From Bayes' theorem, $pr(w \mid s) = pd(s \mid w)ds \times pr(w) \div pd(s)ds$.

The *language model* gives $pr(w)$, the prior probability of the word sequence occurring. The *acoustic model of speech production* gives $pd(s \mid w)$, the probability that the word sequence *w* would generate the observed input speech signal *s*. [2]

(b) The language model expresses the probability of a word sequence $w = w_1, w_2, \cdots$ as

$$pr(w) = \prod_i pr(w_i \mid w_1, w_2, \cdots, w_{i-1}) .$$

In a unigram model, $pr(w_i \mid w_1, w_2, \cdots, w_{i-1}) = f(w_i)$ implying that the probability of any word is independent of the words that preceded it in the sentence.

In a bigram model, $pr(w_i \mid w_1, w_2, \cdots, w_{i-1}) = f(w_i, w_{i-1})$ and in a trigram model $pr(w_i \mid w_1, w_2, \cdots, w_{i-1}) = f(w_i, w_{i-1}, w_{i-2})$. Thus in these models the probability of a word depends on either the previous or the two previous words.

For a 20000 word vocabulary, there are 2×10^4 unigram probabilities, 4×10^8 bigram probabilities and 8×10^{12} trigram probabilities. [3]

(c) Totalling each column to obtain *N*(*j*) gives 81, 65, 102 and 50 respectively for a total of 298 tokens. The unigram probabilities are therefore:

1: 81/298=0.272, 2: 65/298=0.218, 3: 102/298=0.342, "end": 50/298=0.168 [4]

(d) We first calculate *d*(*i*) according to whether or not all valid successors occur 3 or more times. Note that *d*("start")=0 since "end" is not a valid successor of "start".

b(*i*) represents the total probability allocated to the infrequent successors divided by the sum of the corresponding unigram probabilities:

$$b(1) = \frac{1+0+0.5+0.5}{81} \times \frac{1}{0.272+0.218} = 0.0504 \qquad b(2) = \frac{0.5+0.5+2+0.5}{65} \times \frac{1}{0.342} = 0.157$$

We can now complete the table:

p(*i*,*j*)		Second Word, *j*						
		1	2	3	end	N(i)	d(i)	b(i)
Initial Word, *i*	start	0.2	0.2	0.6	0	50	0	
	1	0.014	0.011	0.735	0.241	81	0.5	0.050
	2	0.038	0.762	0.054	0.146	65	0.5	0.15
	3	0.657	0.049	0.098	0.196	102	0	

[8]

(e) Bigram probabilities cannot be estimated accurately for sequences that do not occur in the training data or that occur only very rarely. For these cases, the probabilities are therefore assumed to be proportional to the corresponding unigram probabilities with a scaling constant, *b*(*i*) chosen to ensure that the probabilities of all successors sum to unity.

These rare bigrams are likely to be under-represented in the training data (certainly true for valid bigrams that do not occur at all in the training data).We therefore "steal" some of the probability from the more common bigrams using the discounting factor $d(i)$. The effect of this is to reduce the probability of common bigrams and increase that of rare ones. [3]

6. (a) The best path for $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_t$ with t in state s must have frame t–1 in one of the states, say state i. Since the sub-path $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_{t-1}$ must also be optimum, we must have:

$$B(t,s) = B(t-1,i) \times a_{is} \times d_s(\mathbf{x}_t)$$

Since $B(t,s)$ represents the probability density of the *best* path, we must have:

$$B(t,s) = \max_{1 \le i \le S} \left(B(t-1,i) \times a_{is} \times d_s(\mathbf{x}_t) \right)$$

Since we require frame 1 to be in state 1, $B(1,s) = d_1(\mathbf{x}_1)$ if s=1 and 0 otherwise. [4]

(b) $\ln\left(d_i(\mathbf{x})\right) = -\frac{1}{2}\left(P \ln(2\pi) + \ln\left(|\mathbf{C}_i|\right) + (\mathbf{x} - \mathbf{m}_i)^T \mathbf{C}_i^{-1} (\mathbf{x} - \mathbf{m}_i)\right)$

(i) The use of log probabilities gives two benefits:

Firstly, the dynamic range of the representation is much larger. When many probabilities are multiplied together, the result may underflow the computer's floating point representation.

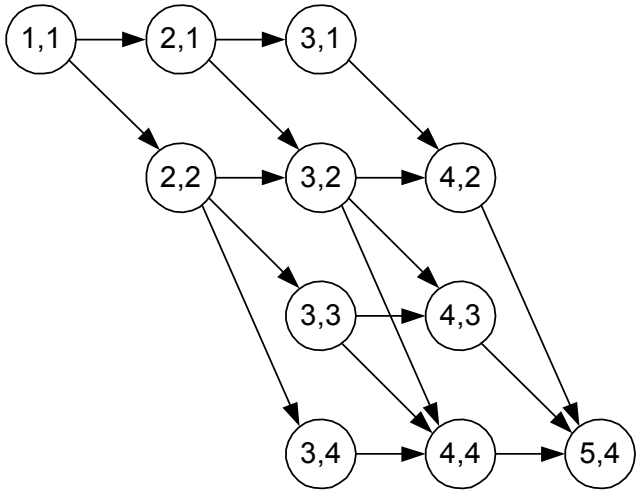
Secondly, the calculation of $d_i(\mathbf{x})$ entails the evaluation of exponentials whereas the evaluation of its log does not. The ln(*) terms in the above expression do not depend on $\mathbf{x}$ and therefore need only be evaluated once.

(ii) Assuming $\mathbf{C}$ is diagonal reduces the number of multiplications needed to evaluate $\ln\left(d_i(\mathbf{x})\right)$ from around $\frac{1}{2}P^2$ to around $2P$. For p=39, this is a saving of around 10 times. [5]

(c) We first make a table of the log transition probabilities:

	1	2	3	4	end
1	-0.511	-0.916			
2		-0.693	-1.204	-1.609	
3			-0.693	-0.693	
4				-0.511	-0.916

Now we can calculate all the B values in the following lattice



B(1,1)	–5	–5
B(2,1)	–5–0.511–9 = –14.511	–14.511
B(2,2)	–5–0.916–5 = –10.916	–10.916
B(3,1)	–14.511–0.511–5 = –20.022	–20.022
B(3,2)	B(2,1): –14.511–0.916–7 = –22.427 **B(2,2): –10.916–0.693–7 = –18.609**	–18.609
B(3,3)	–10.916–1.204–8 = –20.120	–20.120
B(3,4)	–10.916–1.609–8 = –20.526	–20.526
B(4,2)	B(3,1): –20.022–0.916–7 = –27.938 **B(3,2): –18.609–0.693–7 = –26.303**	–26.303
B(4,3)	**B(3,2): –18.609–1.204–8 = –27.813** B(3,3): –20.120–0.693–8 = –28.813	–27.813
B(4,4)	**B(3,2): –18.609–1.609–6 = –26.219** B(3,3): –20.120–0.693–6 = –26.813 B(3,4): –20.526–0.511–6 = –27.037	–26.219
B(5,4)	B(4,2): –26.303–1.609–8 = –35.912 B(4,3): –27.813–0.693–8 = –36.506 **B(4,4): –26.219–0.511–8 = –34.730**	–34.730

[7]

The optimum choice in each case is in bold and corresponds to the path: 1,2,2,4,4.

(d) For Viterbi training, we start with an initial model and iterate the following procedure until it converges:

- Align all the training examples with the model using the above procedure.
- Reestimate $\mathbf{m}_i = \sum \mathbf{x} / N_i$, $\mathbf{C}_i = \sum \mathbf{x}\mathbf{x}^T / N_i - \mathbf{m}_i \mathbf{m}_i^T$ where the sum is taken over all the N_i frames that were aligned to state i.
- Reestimate a_{ij} as the fraction of states aligned with state i for which the next state is aligned with state j.

[4]