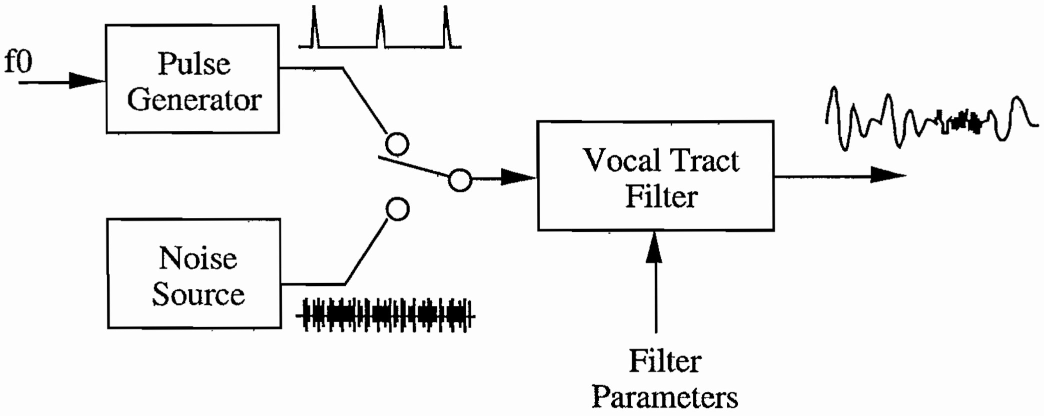
Solutions to 4F11 Speech Processing, 2006

1. *Source-Filter Model and Spectral Block Processing*

   (a) The Vocal Tract Filter is a parametric filter whose transform is intended to model the spectrum of short segments of speech. It is usually an all-pole filter whose pole locations are chosen to approximate the resonances of vocal tract configurations associated with specific speech sounds. The Source or Excitation is switched between two types of excitation. A Pulse Generator produces a periodic signal corresponding to the excitation typical of voiced speech *e.g.* vowels. The period of the pulse train is set based on the pitch of the speech signal. The Noise Source generates a quasi-random signal corresponding to the excitation typical of unvoiced speech. The parameters of the source and filter are updated every 10 msec (or so) in analysis or synthesis of continuous speech.



Source-Filter Model

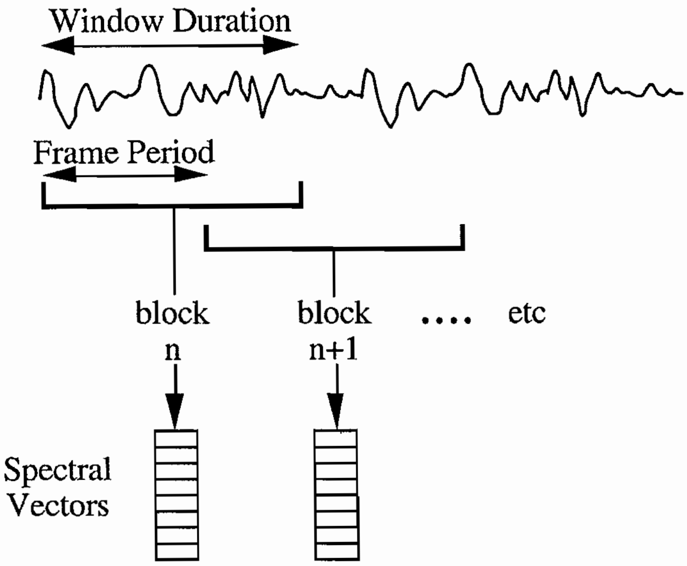
(b) Any two of the following suffice :

(a) The vocal tract can be represented as a single lossless linear time variant filter with a single input.

(b) The excitation is either a periodic pulse train or noise, depending on basic sound classes.

(c) The filter and excitation characteristics are stationary over periods of the order of 10 msecs.

(c)To process continuous speech, a short term spectral estimate must be computed approximately every 10 msecs. Since this is rather a short duration to calculate a spectrum, analysis windows are allowed to overlap so that a longer window can be used, 25 msecs is typical. Each segment of speech is often referred to as a frame. Spectrum Analysis techniques, such as the DFT, are applied to each frame.



Block Processing of Continuous Speech

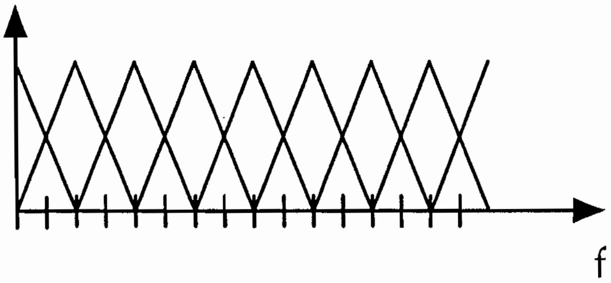
(d)

The DFT produces measurements of spectral energy over a range of frequencies. These frequencies are uniformly spaced over a range determined by the sampling rate. A triangular filterbank can be implemented by applying a series of triangular windows to form a weighted interpolation of the DFT values. The output of each of the triangular filterbank channels represents the energy falling within

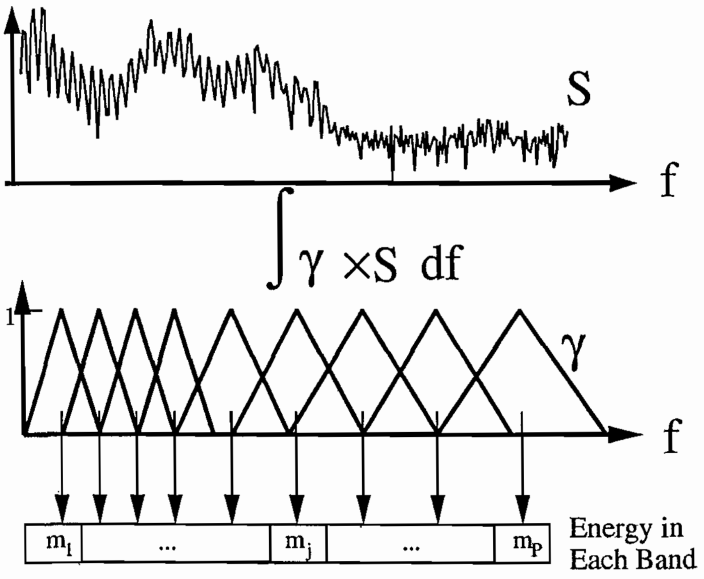
The processing steps are as follows. The speech frames extracted by block processing are windowed, usually by a Hamming or Hanning window. The DFT (or FFT) is applied to each windowed frame of speech to produce the DFT $f_i, i = 1 \ldots N$. The design of the triangular filterbank is determined by the number of channels and the bandwidth of each channel (or equivalently, the overlap between channels). In the accompanying figure, there are 9 filterbank channels which overlap each other by 50%. Each filterbank output receives contribution from 3 DFT bins. The number of DFT frequency measurements is reduced from 16 to 9 channels in the filterbank.

(e) The Mel-scale describes the frequency resolution of human hearing. Small changes in frequency are more perceptible at lower frequencies than at high frequencies.

(f) Rather than being equally space in frequency, the triangular filters are designed so that they are evenly spaced along the Mel-scale. In linear frequency, this corresponds to filters with narrow bandwidth at low frequency and broad bandwidth at higher frequency, as shown in the accompanying figure. The filterbank outputs are computed in the same way as in the uniformly spaced triangular filterbank.



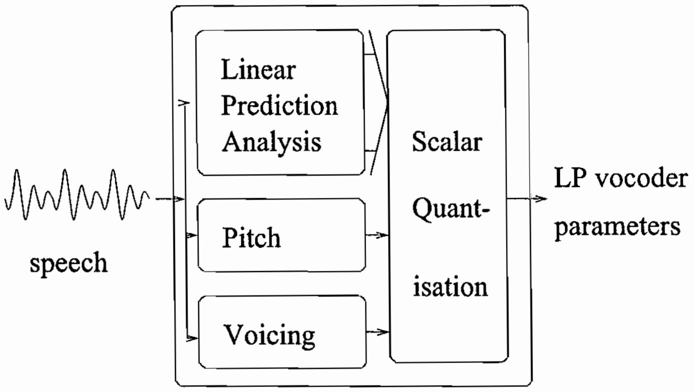
Triangular Filterbank



Computation of a Smoothed Spectrum Using a Mel-scale Triangular Filterbank

2. *LP Vocoder*

(a)



Simple LP Vocoder

(b) For every frame of speech, it is necessary to encode:

- A representation of the LP filter
- Power (to match power of original speech)
- Degree of voicing
- Pitch (if voiced)

Commonly used representations of the LP parameters:

- LP coefficients: when quantisation is not a problem; quantisation errors may lead to unstable LP filters
- Reflection coefficients: can be coded as log area ratios which is an efficient representation; LP filters derived from reflection coefficients are stable even if there are significant quantisation errors

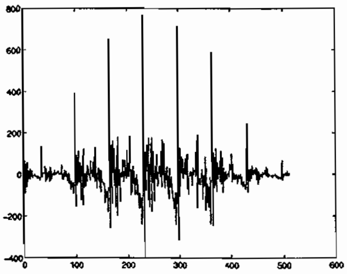
(c) The difficulty arises because the gain, voicing, pitch, and LP filter parameters in the LP vocoder are determined during the analysis of equally spaced frames which have a fixed offset. Parameters obtained from adjacent frames can be interpolated to remove the effect of this discontinuity. It is not feasible to interpolate filter parameters every sample but if done infrequently, parameter updating introduces discontinuities. These discontinuities are less noticeable if parameter updating is synchronised with the pitch excitation.

A practical solution is :

- For unvoiced speech use original analysis timings
- For voiced speech update on pitch pulse injection

- Use linear interpolation to get LP filter (e.g. reflection coefs). Note that direct interpolation of LP filter parameters may lead to unstable filters.

(d)(i)



An Example of The Residual Signal

- In LP analysis, most short term correlations tend to be lost in the error signal. What remains are long-term correlations related to the repeated excitation of the vocal tract by the voicing signal. The residual therefore contains long term correlations due to pitch pulses
- There is a spike at the pitch periods when prediction is poor, owing to the impulse excitation of the vocal tract.

(d)(ii) A simple technique to extract the pitch frequency from the residual is to look for a strong peak in the residual at time lags corresponding to reasonable pitch frequencies, e.g. 20Hz - 40Hz for a male speaker and higher for a female speaker. A more robust method would be to apply autocorrelation analysis to the residual signal itself.

3. *Formants and Format Tracking*

(a) Formants are broad peaks in the speech spectrum that correspond to resonances of the vocal tract.

(b) The first two formants are strong indicators of vowel quality. A low F1 and high F2 is typical of a high front vowel. There is a simple relationship between the tongue and jaw positions, and the values of F1/F2.

	Tongue Front	Tongue Back
High Jaw	F1 Low - F2 High	F1 Low - F2 Low
Low Jaw	F1 High - F2 High	F1 High - F2 Low

(c) Stops are characterised by silence optionally followed by a burst of high energy. The silence is caused by a blocking of the vocal track, by placement of the tongue, teeth, and/or lips. When the blocking is removed, the energy built up behind the obstruction is released as a burst of air. The spectral signature usually consists of a brief period during which little or no energy is visible at any frequency, followed by a brief period during which there is a broad band of energy evenly distributed above a low frequency cut-off, corresponding to the local of the obstruction. Stops may be voiced or unvoiced; in voiced stops, low frequency energy corresponding to the voiced excitation is also visible.

(d) The formant candidates can be chosen as either the N largest peaks or the N lowest frequency peaks in the smoothed linear prediction spectrum. The simplest way to extract peaks is to pick the frequencies at which the peaks are highest. An alternative is to pick frequencies at which the second derivative of the spectrum is the most negative - this yields the sharpest peaks and can also find 'merged' peaks.

(e) The roots of the linear predictor polynomial are useful in estimating formant frequencies because the LP spectrum models the spectral envelope , the peaks of which are the formants.

As shown below, the LP polynomial may be factorised and therefore the method of partial fractions may be used to express the transfer function as:

$$\begin{aligned}\frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}} &= \frac{1}{\prod_{k=1}^{p}(1-z_k z^{-1})} \\ &= \prod_{k=1}^{p} \frac{1}{(1-z_k z^{-1})} = \sum_{k=1}^{p} \frac{C_k}{(1-z_k z^{-1})}\end{aligned}$$

The roots occur in complex conjugate pairs (as the polynomial in a_i is real). The angle of each root defines the pole (candidate formant) frequency. As with the LP spectrum, formants can be chosen either as all peaks in the LP spectrum or those peaks closest to the unit circle , which have the narrowest bandwidth.

(f) A formant tracker imposes constraints upon the formant estimates produced by successive, independent frame-based analysis of the speech spectrum. To obtain a series of formant estimates that is consistent over time, i.e. a formant track, it is possible to impose constraints heuristically upon the measured formants. Some estimates are discarded, and the location of others that are 'missing' can be inferred. For example:

- Restrict range of search for the formant frequencies to 0-3kHz
- If 3 peaks are found, assign to first 3 formants (85% of time for male speech)
- Otherwise assign peaks to formants based on previous frame assignment and distances between peaks and formants. One possibility is simply to replicate a formant found at the previous frame.

This method choses only locally optimal assignments of formant frequencies.

4. *HMM parameter estimation*

(a) Assumptions are that the speech was generated by a first-order HMM of the form described and that the speech is completely represented by the feature vectors. Main assumptions [20%]

- Probability of moving from one state to another (transition probabilities) are fixed and only depend on current state
- Probability density of a frame is independent on all other frames and dependent only on the current state (conditional independence)
- speech vector completely represents frame - includes pseudo-stationary assumption.
- state pdf models true distribution for each state

(b) Backward probability ($\beta_j(t)$) defined as

$$\beta_j(t) = p(\boldsymbol{y}_{t+1} \cdots \boldsymbol{y}_T | s_t = j, \mathcal{M})$$

$\beta_j(t)$ can be computed efficiently

$$\beta_j(t) = \sum_{i=1}^{N} a_{ji} b_i(\boldsymbol{y}_{t+1}) \beta_i(t+1)$$

where $\beta_j(T+1) = 1$ if $j = N+1$ and 0 otherwise. Recursion moves backward in time. [20%]

(c)

$$\begin{aligned}
\alpha_j(t)\beta_j(t) &= p(\boldsymbol{y}_1 \cdots \boldsymbol{y}_t, s_t = j \,|\mathcal{M}) p(\boldsymbol{y}_{t+1} \cdots \boldsymbol{y}_T \,| s_t = j, \mathcal{M}) \\
&= p(\mathbf{Y}, s_t = j \,|\mathcal{M}) \\
&= p(\mathbf{Y} \,|\mathcal{M}) P(s_t = j \,|\mathbf{Y}, \mathcal{M}) \\
&= p(\mathbf{Y} \,|\mathcal{M}) L_j(t)
\end{aligned}$$

Hence, [15%]

$$L_j(t) = \frac{1}{p(\mathbf{Y} \,|\mathcal{M})} \alpha_j(t)\beta_j(t)$$

(d) By considering the GMM as a set of parallel states, can write down the value for $L_{jm}(t)$ [10%]

$$L_{jm}(t) = \frac{1}{p(\mathbf{Y}|\mathcal{M})} \left\{ \sum_{i=1}^{N} \alpha_i(t-1) a_{ij} \right\} c_{jm} b_{jm}(\boldsymbol{y}_t) \beta_j(t)$$

(e) Use [20%]

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_{jm}(t)[(y_t - \hat{\mu}_{jm})(y_t - \hat{\mu}_{jm})']}{\sum_{t=1}^{T} L_{jm}(t)}$$

$$\hat{c}_{jm} = \frac{\sum_{t=1}^{T} L_{jm}(t)}{\sum_{t=1}^{T} L_j(t)}$$

To extend these to multiple observation sequences simply sum over the sequences for both the numerator and denominator of each of the re-estimation formulae. [10%]

5. *Decoding, Context Dependent Models and Language Models*

(a) This is a standard time-synchronous decoding algorithm described in lectures. First compile all the knowledge sources i.e. HMM states for each phone; dictionary pronunciations; and language model (here a unigram) into a single large HMM network. Then find the best (most likely) state sequence through the complete network. To do this use the standard Viterbi algorithm which has the basic equation to extend a path by one frame.

$$\log \Phi_j(t) = \arg\max_i \log \Phi_i(t) + \log a_{ij} + \log b_j(o_t)$$

Note that the unigram language model log probability needs to be added to the accumulated score when leaving each word model. This can actually be added at the start of each word to improve pruning efficiency in a linear lexicon.

It is necessary to record the maximum decisions that are made when leaving each word end at each time step (use word-link record structures in token-passing implementation). When a particular path has been extended to the end of the sentence, the trace back through the WLRs to find the optimal word sequence (if a state-sequence is needed then need to store extra max decisions).

Beam search records the maximum value of $\log \Phi_j(t)$ at each frame and only keeps paths active that are within a threshold of this value. It maintains an active list and only propagates tokens (extends paths) from the states on the active list. Note that after external word-end propagation all word starts will be activated.

The proposed scheme is suitable here, but a tree-based organisation of the lexicon could also be used to improve efficiency. This shares computation very efficiently at the start of the network due to the asymmetry of pruned search. Note that the unigram probabilities will need to be either moved to the end of words (less efficient) or the maximum at each word in the node applied and then the values for each word corrected at the word end nodes. [50%]

Note that more detail, including appropriate network sketches, may be expected from candidates.

(b) (i) Word internal triphones simply replace the models used within each word and don't change the overall structure. Probably not worth using a multi-pass strategy here. [10%]

(b) (ii) Cross-word triphone models require an expansion of the last phone and the first phone of each word (multiple copies) and therefore significantly complicate the network structure. It would definitely be beneficial to use say a word-internal system to generate a lattice or N-Best list before rescoring with cross-word triphones to limit the size of the recognition network. [15%]

(b) (iii) The addition of a bigram language model means that individual word-end to word-begin transitions are needed and that these are weighted by the word-bigram probabilities. Probably not worthwhile having a multi-pass strategy here if a linear lexicon model is used. However if a tree-structured lexicon is used then it may be worthwhile. [10%]

(b) (iv) For trigrams need to expand the network to have an explicit two word history. For 60,000 word vocabulary (depending on the actual number of trigrams present) this is infeasible. Hence some other approach such as multi-pass decoding using e.g. a bigram is required to produce word lattices that can be expanded to trigram lattices and then rescored. [15%]