TR－A－0063

# HMM Speech Recognition using DFT and Auditory Spectrograms

DFTと聴覚スペクトログラムを用いた
HMM音声認識

**Roy D. PATTERSON and Tatsuya HIRAHARA**

パターソン, ロイ D　　平原　達也

1989. 10. 23

ATR視聴覚機構研究所

〒619-02 京都府相楽郡精華町乾谷　☎07749-5-1411

**ATR Auditory and Visual Perception Research Laboratories**

Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1411
Facsimile: +81-7749-5-1408
Telex: 5452-516 ATR J

# HMM Speech Recognition using DFT and Auditory Spectrograms

Roy D. Patterson

MRC Applied Psychology Unit,
Cambridge, England

Tatsuya Hirahara

ATR Auditory and Visual Perception
Research Laboratories,
Kyoto, Japan

1989. 10. 23

# INTRODUCTION

As the performance of speech recognition systems improves, expectations rise and people contemplate using recognition systems in office environments. Unfortunately, the performance of current recognition systems deteriorates badly when they are required to operate in noise -- even office noise. In an attempt to improve performance in noise Ghitza(1988) replaced the traditional Fourier frontend of a speech recognition system with an auditory frontend composed of a bank of auditory filters, a bank of hair cells and an Ensemble-Interval Historgram (EIH) used to summarize the information flowing from the bank of hair cells. It is this final stage that provides most of the noise resistance and gives the auditory model its name, EIH. The recognizer is based on a DTW system described by Wilpon and Rabiner (1985) and it was used to compare the performance of the EIH frontend with the traditional FFT frontend. The results show that in noise free conditions the EIH and FFT systems support essentially the same performance (greater than 90% correct on a word recognition task). However, as the level of the background noise increases, the performance of the FFT system deteriorates more rapidly than that of the EIH system. In the case of male speakers the advantage of the EIH system in noise is dramatic; in the case of female speakers, however, the superiority of the EIH system is marginal.

In this paper we describe a similar attempt to demonstrate the advantage of an auditory frontend for a recognition system that has to operate in noise. Instead of the EIH auditory model, we use an Auditory Sensation Processor (ASP) that simulates the auditory images that we experience in response to music and speech sounds. The architecture of ASP is similar to that of EIH, inasmuch as it has three stages -- a filterbank, a 'haircell bank' and a 'neural processor' that removes noise in the time domain using a correlation process, but ASP has several potential advantages. Firstly, the haircell stage of the ASP model includes lateral suppression which sharpens features in the output of the filterbank, and so it might be expected to improve the

performance in noise over that provided by the EIH model. Secondly, although the 'neural processor' in ASP has the same function as that in EIH, it uses a data driven mechanism that is simpler than autocorrelation and so the ASP frontend is probably faster. Finally, the output of the ASP model is very similar to the traditional spectrogram and so it is easier to read than EIH output and it can be connected directly to recognition systems designed to work with spectrographic input.

The speech recognizer was an HMM system described in Waibel, Hanazawa, Hinton, Shikano and Lang (1988). It was designed to take spectrographic input and its performance on a syllable spotting task is well documented. In the present study, the speech stimuli used by Waibel et al(1988) were converted to spectrograms using both the traditional DFT procedure and an auditory model referred to as ASP. In one condition the speech was noise free and in the other a loud pink noise was added to the speech sounds. The DFT and SAS systems were trained separately with the clean speech and the noisy speech using half of the syllable database. Then, they were tested on the other half of the data base using both clean speech and noisy speech. This procedure enabled us to test the ability of the two recognizer systems to generalize what is learned from one form of the speech (clean or noisy) to the other (noisy or clean) -- a particularly relevant form of generalization for a practical recognizer.

The first section of this paper describes ASP and the tuning of the model for use with speech in noise. The second section describes the results of the recognition tests.

# I. AUDITORY SENSATION PROCESSING

The term Auditory Sensation Processing (ASP) is meant to describe all of the mechanical and neural processing applied by the auditory system to a sound to construct the sensation, or the initial auditory image, that we hear in response to that sound. A schematic representation of sensation processing and its assumed place within audition is presented in Figure I.1. The box at the top of the figure represents all of audition. By this we mean all of the signal processing necessary to recognize individual words in speech, when those words occur in an unambiguous context. The second row shows that we think of this auditory processing as composed of two main sets of subprocesses -- those that convert the waveform coming from the air into basic auditory sensations, and those that convert the sensations into speech perceptions. Auditory sensation processing itself is presented in the lower half or the figure, and it is also divided into two parts -- peripheral and central. By 'peripheral' we mean the operations performed in the cochlea, or inner ear. By 'central' we mean the processing required to convert the output of the cochlea into the sensations that we hear when presented with a particular sound.

In the auditory system, the peripheral processing begins with a spectral analysis performed by the basilar membrane in conjunction with the outer hair cells. In the ASP model, the spectral analysis is performed by a gammatone auditory filterbank which converts the incoming wave into a surface that provides a reasonable representation of the motion of the basilar membrane as a function of time. In the auditory system, the inner hair cells convert the stimulus into neural transmitter whose concentration determines the probability of firing for the sensory nerve fibres to which they are attached. The process includes temporal adaptation and lateral suppression which would appear to enhance features that arise in the basilar membrane motion.^A This suggests that the bank of haircells should be regarded as a sophisticated signal processor rather than just a neural transducer. In the ASP model, the operation of the inner hair cells is simulated by a module that includes a bank of logarithmic

compressors and a bank of adaptive threshold generators (Holdsworth, 1989). Together they convert the basilar membrane motion into a surface that represents the pattern of neural activity at the output of the cochlea. The adaptive thresholding mechanism removes the temporal and spectral smearing introduced by the filterbank and it enhances features in the filterbank output. The gammatone filterbank and adaptive thresholding are described in Sub-sections I.A. and I.B, respectively.

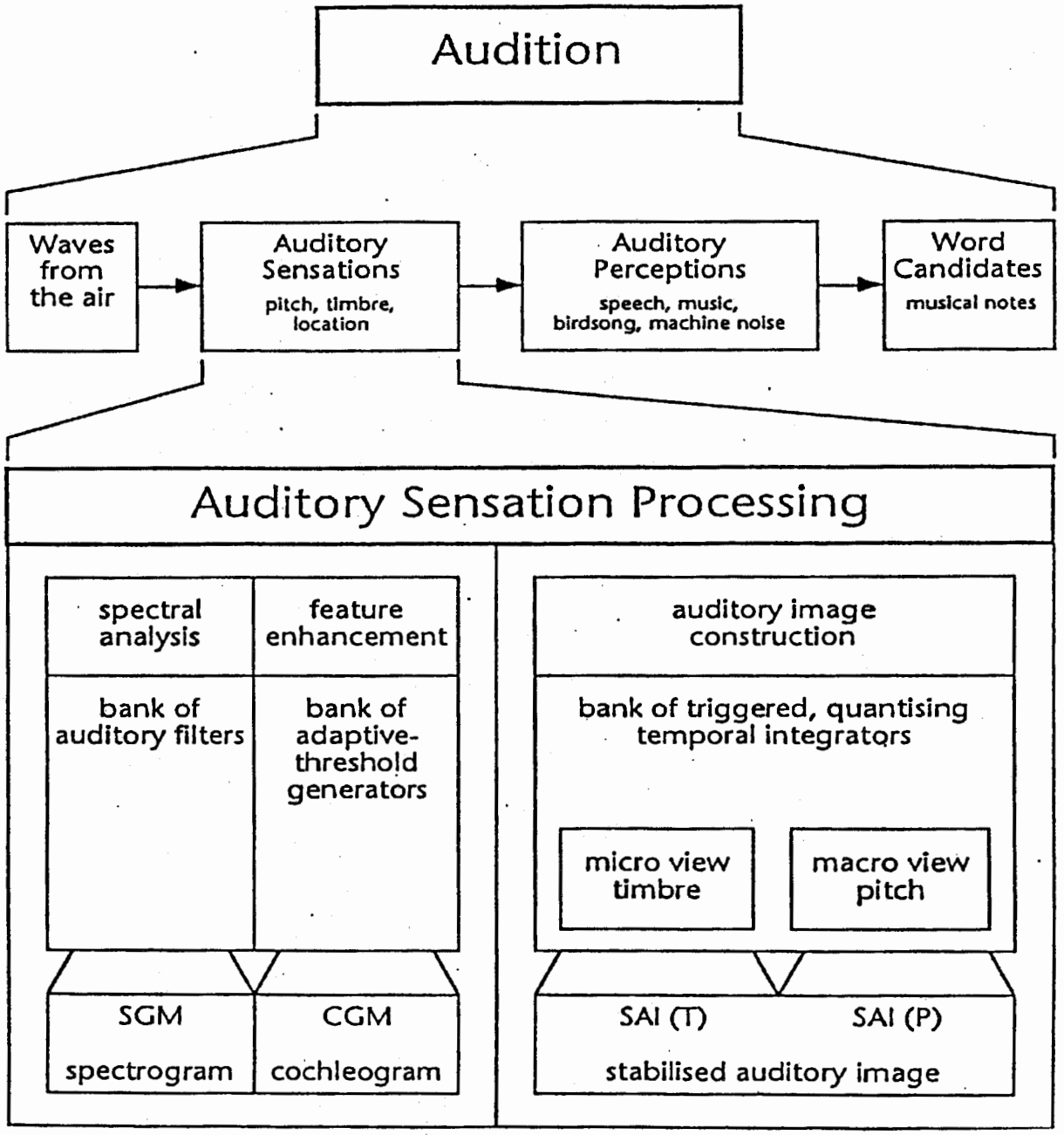


Figure I.1 The structure of the computational version of ASP.

In the auditory system, the output of the cochlea proceeds through a sequence of brain stem and mid-brain nuclei to the auditory cortex. As yet we have little physiological information concerning the processing performed along this path or in the cortex. What we do know, however, from introspection and psychological experiments, is auditory image. When the incoming sound is periodic the auditory image is stationary with fixed pitch and timbre. The purpose of the 'central' part of the ASP model is to convert the output of the cochlea into something like our auditory image, that is, a visual display, or other representation, that is stationary when the sound is stationary and which only changes when we hear the sound change. The image construction process is a form of triggered, quantized temporal integration; it stabilizes periodic sound components and it increases the contrast of periodic sound components at the expense of aperiodic sound components. The voiced parts of speech are quasi-periodic sounds that should benefit from the signal enhancement provided by the image construction process. The mechanism is described in Sub-section I.C.

In summary, with regard to speech recognition systems, ASP would appear to have four advantages over traditional frontends. The adaptive thresholding mechanism used to simulate the operation of the inner haircells removes the smearing introduced by the filterbank while performing the spectral analysis and it sharpens features in the output of the filterbank. The temporal integration process used to construct the auditory image stabilizes formant information over glottal cycles and increases the contrast between voiced speech features and noise.

## A. Spectral Analysis: The Gammatone Auditory Filterbank

In ASP the spectral analysis is performed by a gammatone auditory filterbank. The motivation for adopting the gammatone filter shape is threefold: (1) It provides an excellent summary of the physiological data concerning the frequency response and the temporal response of primary auditory neurons in small mammals such as cats (Carney, 1988). (2) When adapted to human

parameter values it provides an excellent representation of the frequency response of the human auditory filter, and so it predicts auditory masking well (Patterson et al, 1988). (3) We have discovered a recursive implementation of the gammatone filter that makes it particularly fast (Patterson et al, 1988). The parameters for the filterbank and the values used in this study are shown in Table 1.

| Option Name | Value | Brief Description |
|---|---|---|
| mincf_afb | 200 | Minimum center frequency (Hz) |
| maxcf_afb | 4000 | Maximum center frequency (Hz) |
| dencf_afb | 1.5 | Filter density (filters/critical band) |
| bw_gtf | 1 | Filter bandwidth scalar |
| order_gtf | 4 | Filter order |

Table 1. The parameters for the Gammatone Auditory Filterbank.

The top group with the common suffix afb (Auditory FilterBank) control the distribution of the filters across frequency and the total number of filters in the bank. Mincf_afb is the minimum center frequency below which there are no filters; maxcf_afb is the maximum center frequency above which there are no filters; and dencf_afb is the filter density. When dencf_afb is one, the filter centers are separated by one Equivalent Rectangular Bandwidth (ERB). The ERB is about 14% larger than the 3 dB bandwidth of the filter, and the ERB values are those for young normal listeners, taken from the equation of Moore and Glasberg (1983). With this combination of parameter values there are 32 filters in the bank and 32 channels in the stabilized auditory spectrogram.

The impulse response of the gammatone filter is

$$Gt(t) = a \cdot t^{(n-1)} \cdot \exp(-2\pi bt) \cdot \cos(2\pi f_o t) \quad (t >= 0)$$

where a is a scalar, n is the filter order, b determines the bandwidth and fo is the center frequency of the filter. The term 'gammatone' refers to the fact that the envelope of the impulse response of the filter (the expression in square brackets) is the traditional gamma function from statistics and the fine-structure (the cosine term) is a sinusoid, or tone, at the center frequency of the filter (de Boer, 1988).

In the current version of ASP, the bandwidths are not set individually; rather, the values are taken from the critical band function for young, normal adults (Moore and Glasberg, 1983). The bandwidth parameter, bw_gtf simply increases or decreases all of the bandwidths by the same proportion; in this study it was fixed at 1.0. The order of the filter, order_gtf, is the number of filtering stages. It determines the slope of the skirts of the attenuation function and their extent, but it has little effect on the passband of the filter for orders greater than three. The value used in this study was 4.

The output of the filterbank in response to a small segment of the vowel in the demi-syllable 'ba' is shown in Figure I.2. Each of the fine lines in the figure shows the output of an individual auditory filter. Together the set of filter outputs define a surface which represents the motion of the basilar membrane as a function of time in response to this stimulus. Each time a glottal pulse strikes the resonators of the vocal tract, it produces concentrations of sound energy that appear as auditory features in the basilar membrane motion; sequences of these features are referred to as 'formants'. The first formant appears about a third of the way up the figure as a pair of relatively strong harmonics that are largely resolved. The remaining formants (the second, third and fourth) appear as streams of triangular features in the upper half of the figure. As the formant number increases, the triangles become shorter in time and broader in frequency. Vowel distinctions are determined by the position, strength and shape of the formants.

The surface in Figure I.2 illustrates the basic properties of basilar membrane motion. In the high-frequency channels where the filters are broad, the glottal pulses generate a sequence of impulse responses, each of which dies away before the next glottal pulse occurs. The impulse response in the center of the formant where the sound is most intense dies away last. As the center frequency decreases, the filter bandwidth decreases and the impulse response gets longer. Eventually, it reaches a point where the filter is still ringing when the next pulse arrives. In the lowest channels the filters isolate individual harmonics of the pulse train and the wave at the output of the filter is sinusoidal in shape.

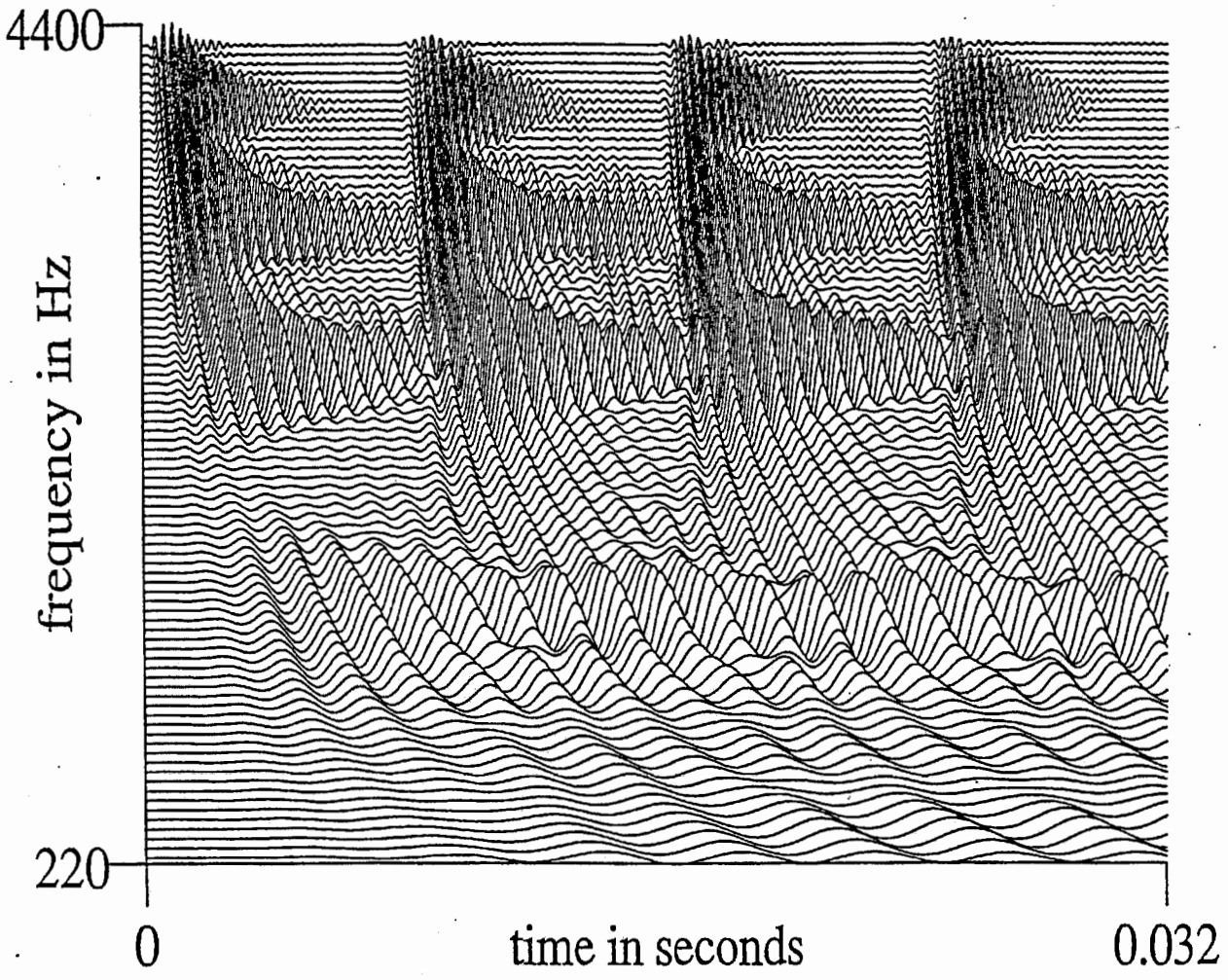


Figure I.2 The response of the gammatone filterbank to vowel /a/.

In the study comparing HMM and TDNN phoneme recognition by Waibel, Hanazawa, Hinton, Shikano and Lang (1988), the phoneme tokens were excised from recordings of continuous speech. Each token was 160 ms in duration and it was centered on the vowel onset. A 256-point DFT was used to compute a sixteen-channel, mel-scale spectrogram. The analysis window was 21.5 ms in duration and between frames the window was stepped forward 5 ms. Adjacent frames were then averaged, and so each phoneme token is represented by a 16-channel, 15 frame spectrogram in which each frame represents 10-ms of the original sound. In an effort to produce results that could be compared with this and other studies at ATR, the same DFT analysis was used in the present study.

From the point of view of auditory perception, the resolution of a 16-channel spectrogram with 10-ms frames is rather limited, and it is possible that it is insufficient resolution to reveal the advantages of an auditory frontend. However, at this point in time, recognition systems cannot accept input data rates much higher than this if they are to support real-time speech recognition. Accordingly, we took the data rate implied by the 16 by 15 spectrogram as a constraint for the auditory frontend and considered the best way to distribute the resources.

With regard to the distribution of channels across frequency, it takes 27 channels to span the frequency range 100 to 5000 Hz with a filter density of unity. If the speech features (particularly formants) are narrow with regard to filter density, the feature enhancement mechanism cannot sharpen them effectively. Ideally, it requires a filter density of 3 or more, that is, a minimum of 80 filters. The question, then, is how to reduce the ideal to 16 channels.

We began by noting that in speech sounds, the information in the region below 200 Hz does not warrant the number of channels that an auditory model assigns it. The only information concerns the presence or absence of the fundamental of any voiced sounds and this information is almost invariably duplicated

in the channels associated with the second harmonic. Accordingly the minimum center frequency, mincf_afb, was increased to 200 Hz. Similarly, there is little speech information in the region between 4.0 and 5.0 kHz. This region is useful for detecting and distinguishing the phonemes /s/ and /f/ in English, but it is not crucial for the BDG phoneme task used in the current study. Accordingly the the maximum center frequency, maxcf_afb, was reduced to 4.0 kHz.

With a filter density of 3.0, the ASP model requires 64 filters to cover the frequency range from 200 to 4000 Hz. This is four times less than the number of points in the DFT (256), but still four times more than the number of channels in the ultimate spectrogram (16). We considered three methods of reducing the frequency resolution to 16 channels:

a) Analyze with 64 channels (dencf_afb=3.0) and reduce to sixteen channels by averaging adjacent sets of four channels.

b) Analyze with 32 channels (dencf_afb=1.5) and reduce to sixteen channels by averaging adjacent pairs of channels.

c) Analyze with 16 channels (dencf_afb=0.75) and increase the filter bandwidth scalar (bw_afb) from 1.0 to 1.33 to ensure that components do not fall between filters.

Figure I.3 presents four spectrograms of one token, ba.30, to illustrate the data used in making the decision. Figure I.3d is a high resolution spectrogram with 64-channel frequency resolution and 1.25-ms temporal resolution which is included to show the resolution available from the auditory model. The remaining sub-figures (I.3a, I.3b and I.3c) show the 16-channel spectrograms that result from the three reduction schemes outlined above. A comparison of Figures I.d and I.c indicates that much of the detail is lost when a 16-channel filterbank is used to produce the spectrograms directly without any subsequent averaging (option a). A much better spectrogram is produced with option (b), where a 32-channel filterbank and pair-wise averaging are employed

(compare Figures I.3b and I.3c). Option (a), a 64-channel filterbank with 4-way averaging produces a further increase in the resolution of the 16-channel spectrogram. However, it doubles the computation time and the additional resolution was judged to be insufficient to warrant the additional computation. These and other comparisons led us to choose option (b), the 32-channel filterbank with pair-waise averaging of channels.

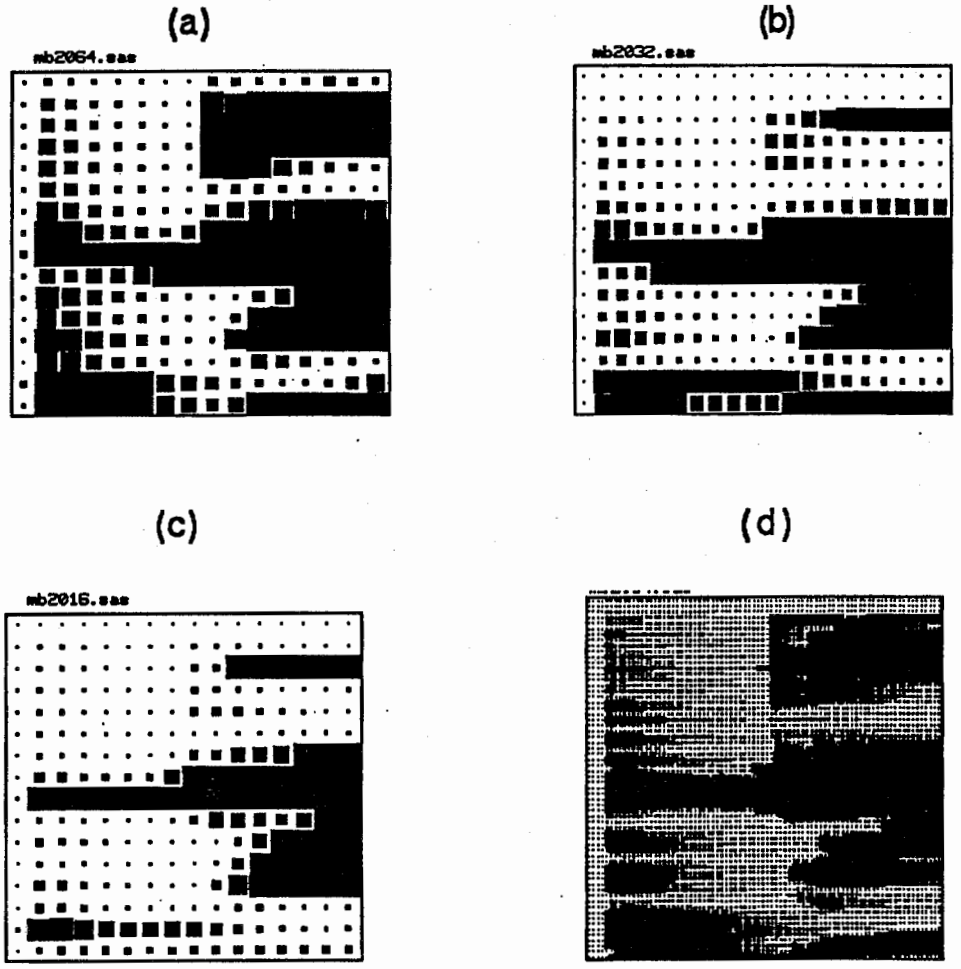


Figure I.3 Four Stabilized Auditory Spectrograms of one token **ba.30**.

(a) 16 by 16 spectrogram from 64 by 120 spectrogram.
(b) 16 by 16 spectrogram from 32 by 120 spectrogram.
(c) 16 by 16 spectrogram from 16 by 120 spectrogram.
(d) 64 channel by 120 time bins' original spectrogram.

B. Adaptive Thresholding

In the auditory system, the bank of inner haircells mounted along the edge of the basilar membrane transduce the mechanical energy of the basilar membrane into neural transmitter which ultimately generates the pattern of firing in the auditory nerve. The haircells compress and rectify the basilar membrane motion. They adapt rapidly to changes in the overall level, and the channels interact in the frequency dimension so that larger features tend to suppress smaller features. In the current version of ASP these four processes -- compression, rectification, adaptation and suppression -- are combined into one module that simulates inner haircell processing. The input to the module is the simulated basilar membrane motion flowing from the auditory filterbank and the output of the module is the 'cochleogram' which is the ASP representation of the pattern of neural activity produced by a sound in the auditory nerve.

The output of each auditory filter is compressed separately and, in the current system, the compressor is strictly logarithmic. In the auditory system, the compressor is logarithmic over the central part of its range and then it asymptotes to a soft limit. When compression is applied to the filterbank response to the vowel [a] shown in Figure I.2, the result is the compressed vowel shown in Figure I.4. Since the compression is logarithmic, the non-positive values of the filtered outputs are set to a small positive number with the result that the filterbank output is half-wave rectified. Compression is required both in the auditory system and in an auditory model because of the enormous dynamic range of the first stage; without it small features that we hear would simply be lost. Unfortunately, compression produces a reduction in the contrast of the features in the filterbank output; that is, the formants are less well-defined in this representation.

In order to reintroduce, and perhaps enhance, the contrast of the features, an adaptive thresholding mechanism is applied to the filterbank output. Threshold values are maintained for each channel and updated at the sampling rate. The new value at any instant is determined by one of four levels: the prior activity in that channel, the prior activity in the channel above, the prior

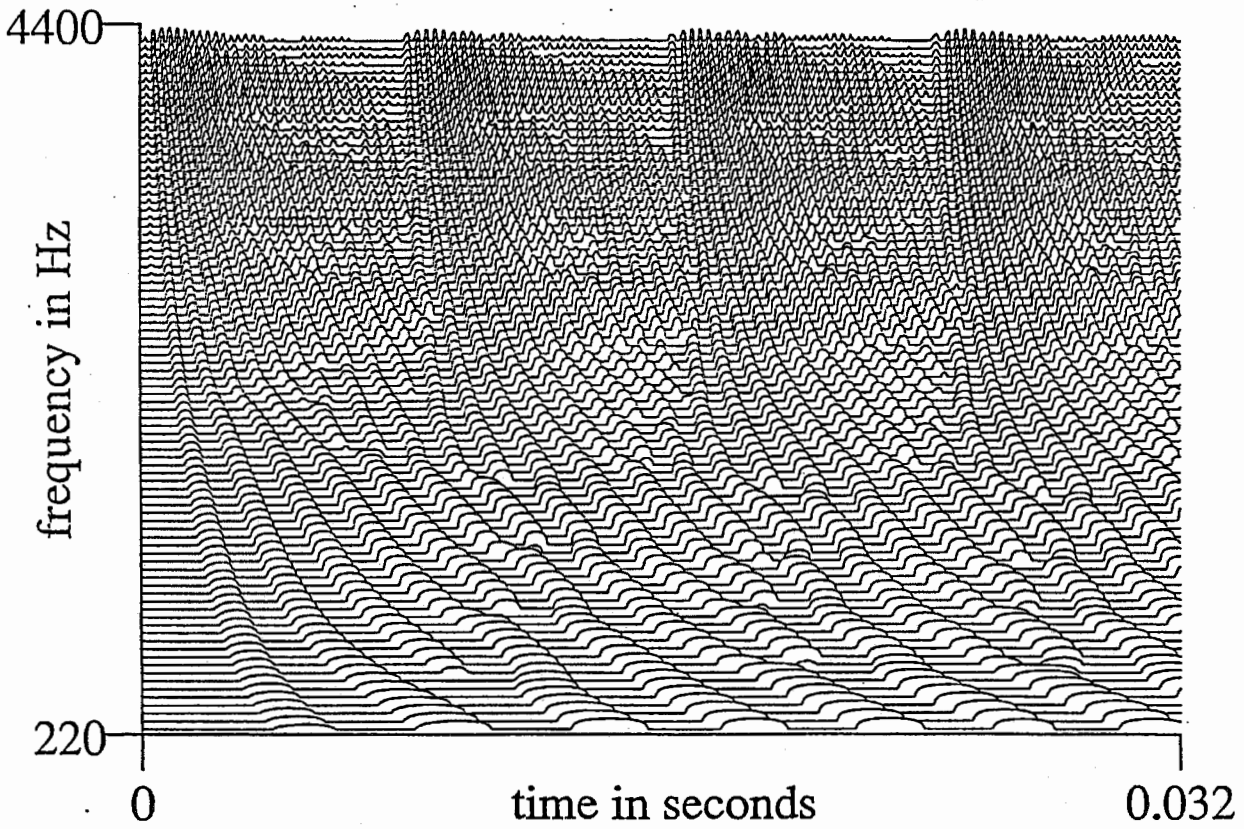


Figure I.4 The compressed filter output to a vowel /a/.

activity in the channel below, or a fixed floor level. The mechanism produces output when the input exceeds this rapidly adapting local threshold. Since intense activity in one channel flows into neighboring channels with lower activity levels, the mechanism is referred to as 'two-dimensional adaptive thresholding'. The mechanism is controlled by four parameters which are shown in Table 2 along with their current values.

| Parameter | Value | Brief Description |
|---|---|---|
| trise_at | 10000 | Threshold rise rate |
| trecovery_at | 0.25 | Recovery rate relative to filter |
| frecovery_at | 5000 | Recovery rate across frequency |
| reclimit_at | 5 | Limitation on recovery level |

Table 2. The parameters for two-dimensional adaptive thresholding

The parameter trise_at specifies the rate at which the adaptive threshold will rise in response to a rise in signal level. It has been set to a value which essentially causes the threshold to follow the envelope of any rise in signal amplitude. The parameter trecovery_at determines the rate of decay of the adaptive threshold relative to the rate of decay of the auditory filter in the absence of further input, that is, the natural temporal response of the filter. Values of trecovery_at less than unity cause the adaptive threshold to decay more slowly than the auditory filter and thereby to remove the filter's temporal response from the representation. This produces an effect that is similar to short term adaptation in the system. The parameter frecovery_at specifies the rate at which the threshold value in one channel propagates to influence threshold in neighboring channels, relative to the natural spread of energy across channels in a filterbank. Values greater than 1000 cause the adaptive threshold to decay in frequency more slowly than the natural spread across channels. This produces and effect that is similar to suppression; a high level of activity in one channel maintains elevated thresholds in neighboring channels and so prevents them from responding to weak signals in those neighboring channels.

In order to prevent the mechanism from encountering system noise, or alternately, to reduce sensitivity to stimulus noise, there is a limit placed on the recovery that the adaptive threshold can achieve. The limit, reclimit_at, is the limit of the sensitivity of the system.

When the parameters are set to the values shown above and the input is the vowel [a], the result is the cochleogram shown in Figure I.5. The adaptive thresholding mechanism restores, and even improves, the contrast of the formants of the vowel. The effect of trecovery_at is illustrated in Figure I.6 where the value has been increased to 0.5. The threshold decays more rapidly and the mechanism detects more activity. Thus, trecovery_at controls short-term adaptation beyond that required to remove the filter response, and effectively produces temporal suppression. The tuning of the suppression mechanism for use with speech sounds was performed with the complete model and so it is illustrated at the end of the next sub-section.

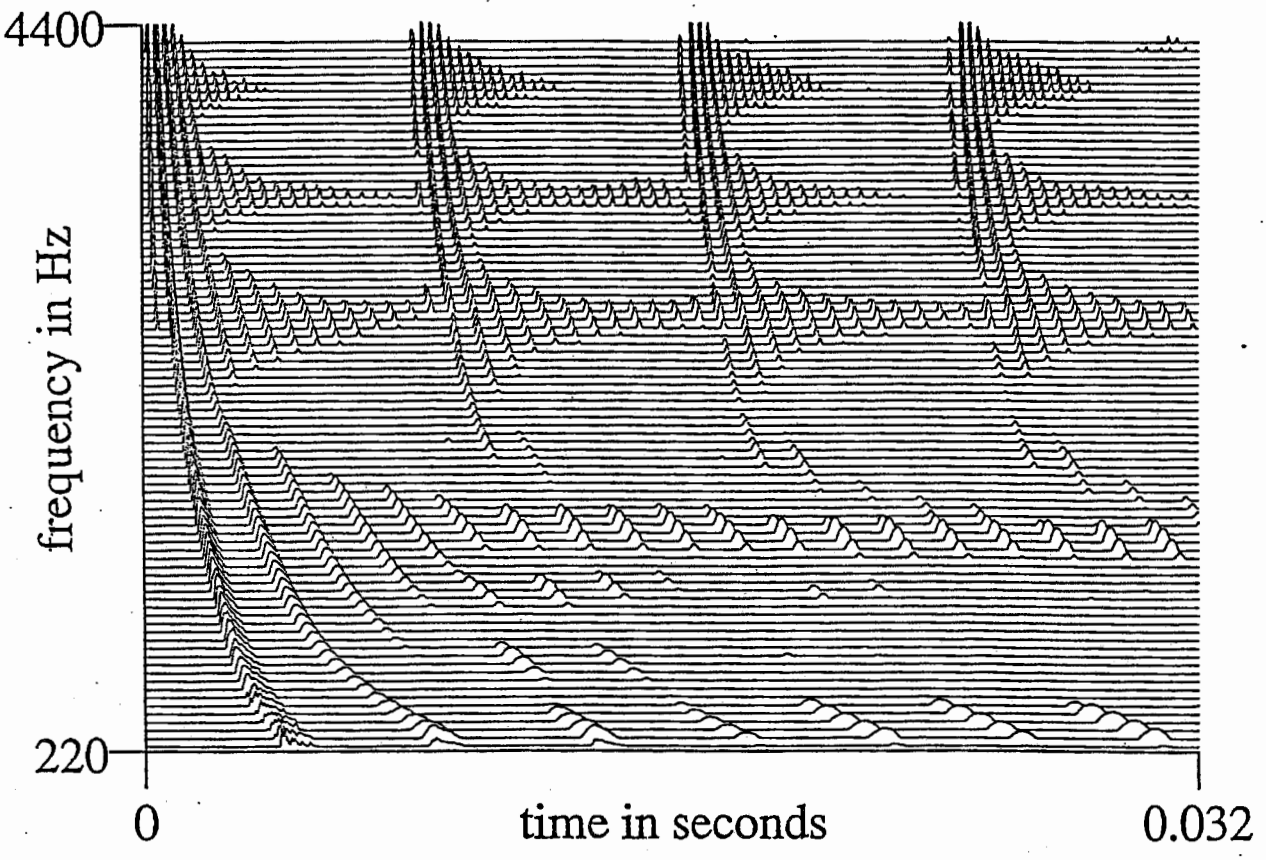


Figure I.5 The cochleogram of a vowel /a/ when **trecovery_at** = 0.25

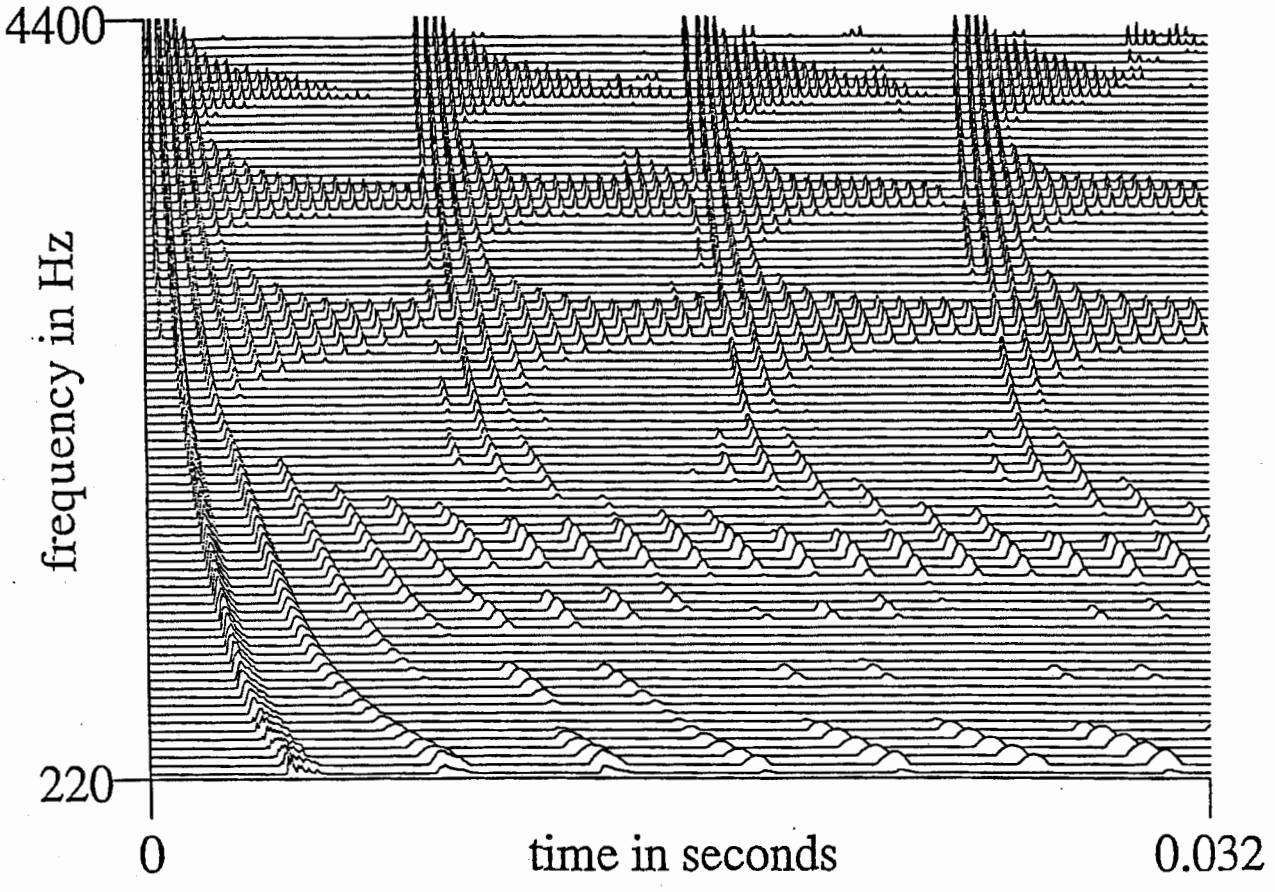


Figure I.6 The cochleogram of a vowel /a/ when **trecovery_at** = 0.5.

C. The Stabilized Auditory Image

When the input to the cochlea is a periodic sound, like a vowel or a musical note, the output oscillates. In contrast, the sensation produced by such a sound does not flutter or flicker; indeed, periodic sounds produce the most stable auditory images. As a possible solution to this discrepancy it has been suggested that we integrate the cochleogram over time, and so smooth out the rapid oscillations of the periodic sound, to produce a stable central spectrum that forms the basis of our stable auditory image. Unfortunately, this cannot be the case because there is evidence to show that fine-grain temporal information in the cochleogram is preserved in the auditory image; fine-grain information that would be integrated out in any simple integration process. Thus the problem in modelling temporal integration is to determine how the auditory system can integrate information over 10 to 100 cycles of a periodic sound without losing the fine-grain temporal detail within the individual cycles of the cochleogram.

In the ASP model the larger peaks in the cochleogram are used to trigger a quantized temporal integration process. The triggering mechanism identifies the individual cycles of periodic sounds and enables us to perform period-synchronous integration in a way that causes periodic information to accumulate and aperiodic information to die away. At the same time, the periodic information forms a stabilized auditory image (SAI) that provides a reasonable representation of the sensation that we hear.

The tuning was done with a tone pip presented repeatedly in a continuous background noise. The duration of the pip was 50 ms and it had 5 ms onset and offset ramps. The tone frequency was 1.0 kHz. The SAI for the noise on its own is presented in the lower part of Figure I.7; the minimum and maximum center frequencies in the figure are an octave below (500 Hz) and an octave above (2000 Hz) the tone frequency. In the center of the figure where the tone will appear, there is a drifting noise component whose phase lag increases across the signal region.

The upper part of the figure shows the SAI after the signal has been on for about 30 ms. The noise in the signal channels is completely suppressed and it is largely repressed in the half-octave regions adjacent to the signal. Outside this region the noise level is largely unchanged. The figure illustrates how the signal contrast builds up in the SAI and the noise suppressed in the ASP model. An extended study of the joint effects of temporal suppression (trecovery_at) and frequency suppression (frecovery_at) led to the conclusion that there is abroad plateau of values in the region where trecovery_at is between 0.125 and 0.5 and frecovery_at is between 2500 and 10000 where the two parameters trade off to produce roughly comparable noise suppression. Accordingly, we chose values in the center of the plateau with trecovery_at set to 0.25 and frecovery_at set to 5000.

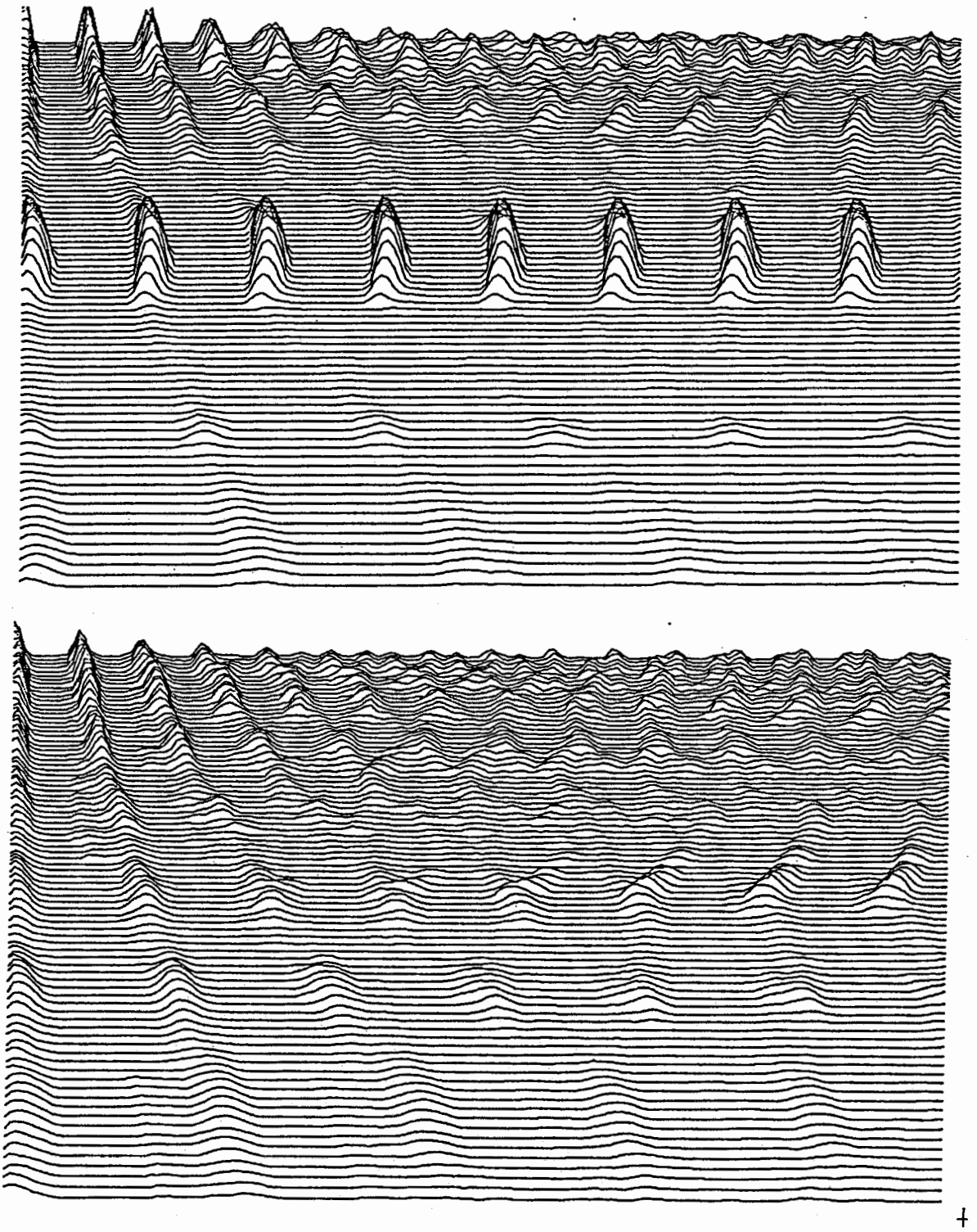

Figure I.7 The upper figure shows the Stabilized Auditory Image for 1kHz pure tone with white noise.The lower figure shows the SAI for white noise alone.

## II. RESULTS

The results for the DFT and SAS recognition systems are shown in Figures II.1 and II.2, respectively. The abscissa is 'codebook size', that is, the number of reference vectors provided to encode each category of stimulus, /b/, /d/, or /g/, in the training set. The ordinate is the percent correct identification of the individual spectrograms (/b/, /d/, or /g/) for a particular combination of training condition and test condition.

### A. Direct Comparison of DFT and SAS Systems

Consider first the relative performance of the DFT and SAS frontends: The results for the two preprocessors have the same general form, in the sense that performance deteriorates as codebook size decreases, and performance in noise is worse than performance on clean speech. Furthermore, the absolute level of performance is the same when the codebook size is large (85). However, as codebook size decreases, performance deteriorates more slowly in the case of the SAS frontend, and the decrement caused by the introduction of noise is smaller in the case of the SAS.

The advantages of the SAS system are most easily observed in Figure I.3 which presents a comparison of the DFT and SAS results for conditions where the system was trained and tested on clean speech (upper four curves) and for conditions where the system was trained and tested on noisy speech (lower four curves). When the stimuli are clean speech and the codebook is large (40 or 85), the performance of the two frontends is essentially the same. But when the system is required to use a smaller codebook performance is better for the SAS frontend. When the speech stimuli are presented in noise, there is a large performance decrement in all cases, but the decrement is larger for stimuli processed through the DFT frontend, and once again, the difference between the two frontends grows as the codebook size decreases. Together these results suggest that the SAS

frontend extracts more general characteristics of the stimuli, particularly when the size of the recognition system is limited.

Part of the SAS advantage is probably the result of the suppression mechanism which enhances the speech features at the expense of the noise whenever the speech is more intense than the noise. At the same time, however, the suppression mechanism reduces small speech features, and in some cases it eliminates them from the record altogether. It is probably this reduction of small features that eventually limits the performance of the SAS system and prevents it from achieving higher levels for clean speech when the codebook size is large. This suggests that we should reduce the level of suppression somewhat in hopes of preserving more speech features while still maintaining the feature enhancement and noise resistance provided by the suppression mechanism.

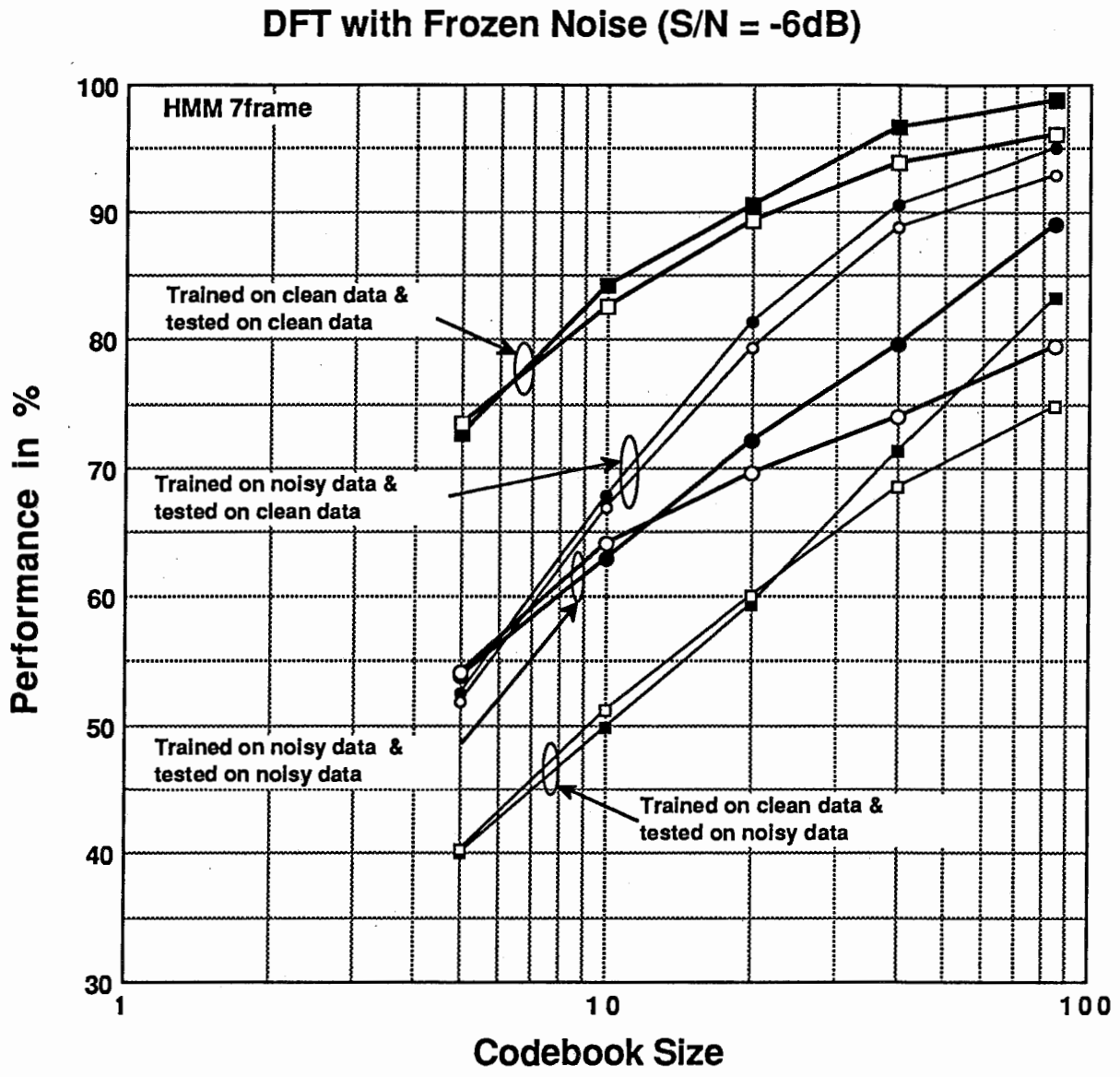


Figure II.1 Results for the DFT. Open symbols and filled symbols indicate results for the open and the closed recognition experiments.

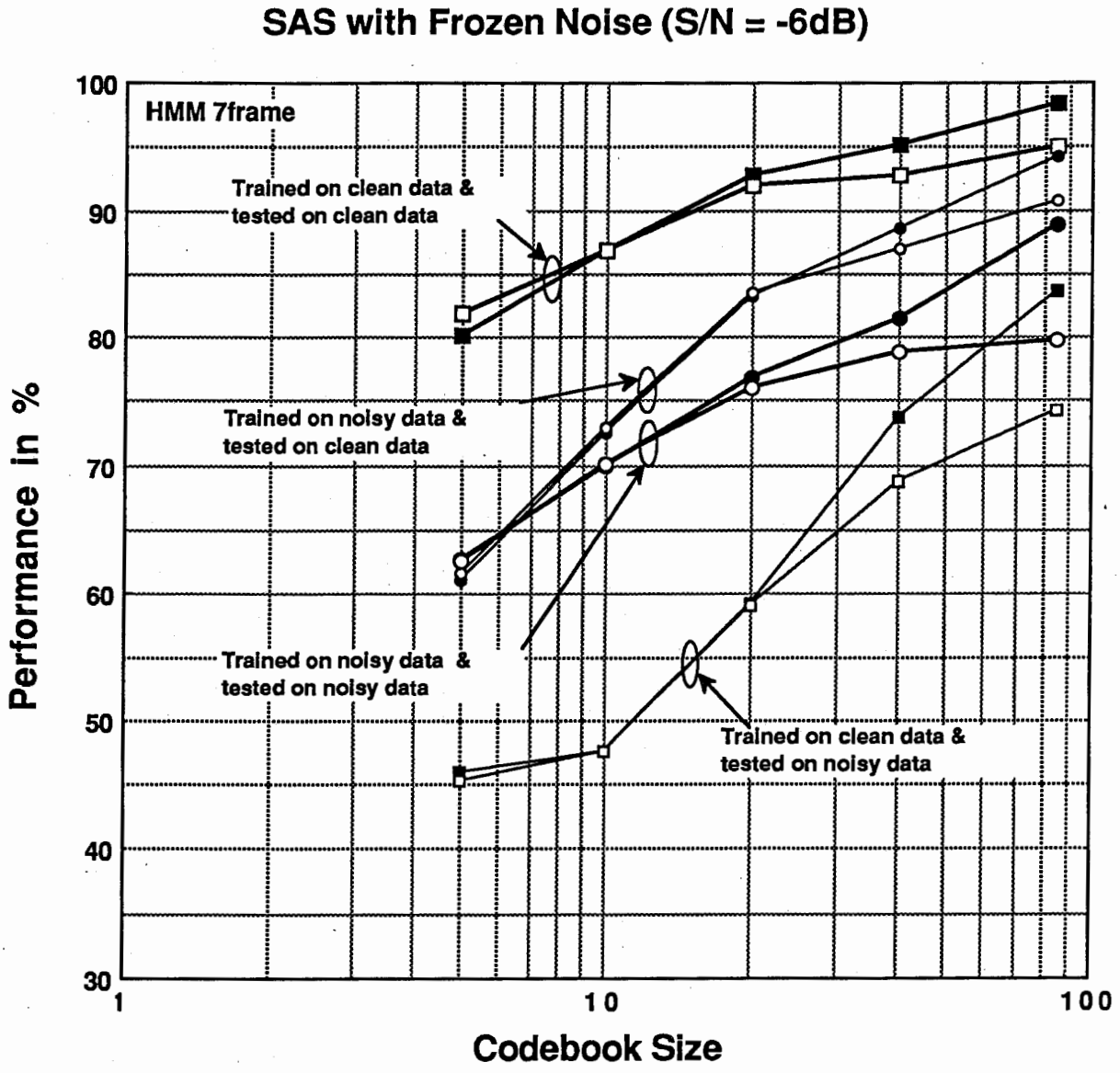


Figure II.2 Results for the SAS. Open symbols and filled symbols indicate results for the open and the closed recognition experiments.

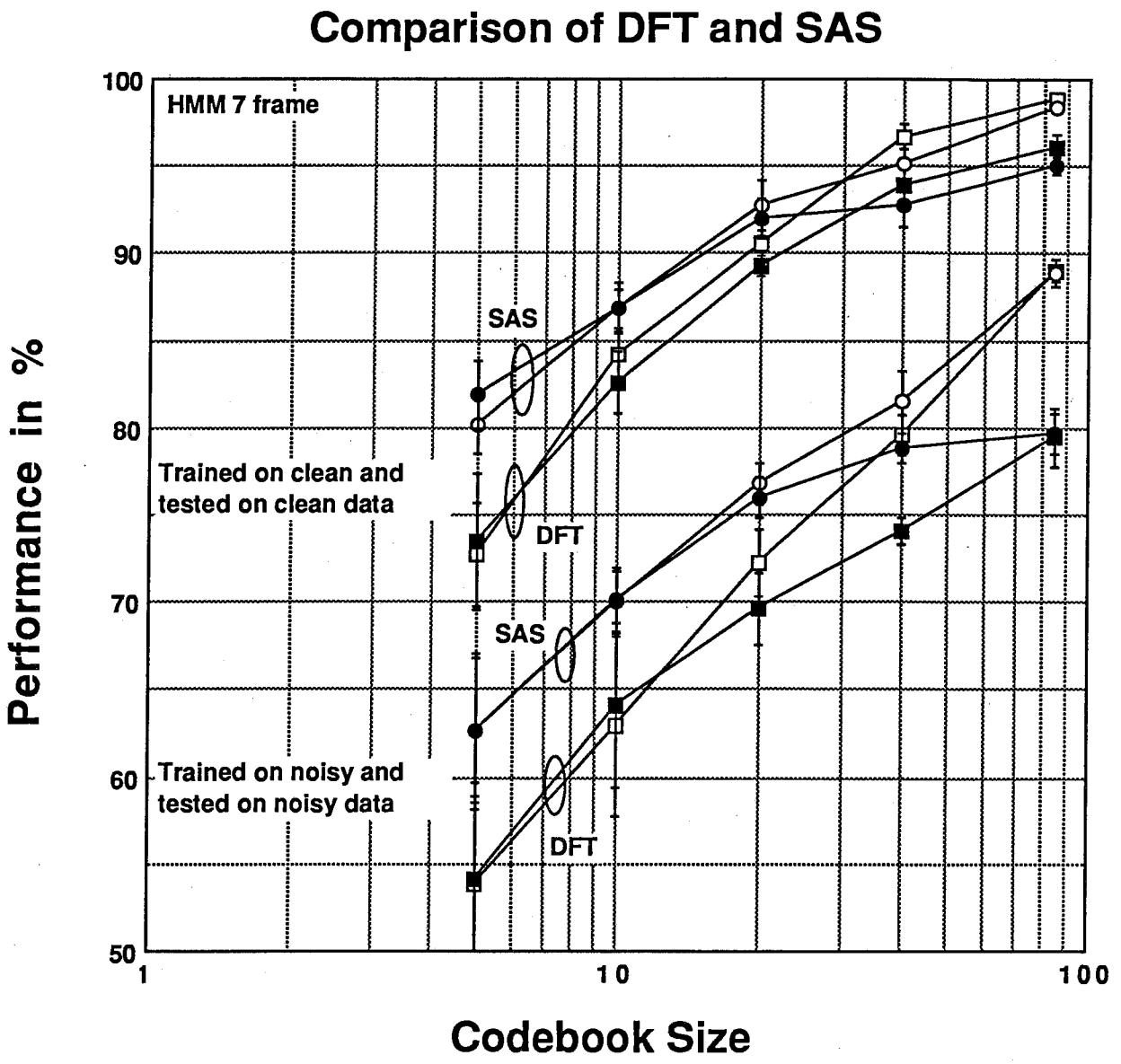


Figure II.3 A comparison of the DFT and SAS results for conditions where the system was trained and tested on clean speech(upper four curves) and for the conditions where the system was trained and tested on noisy speech (lower four curves).

B. Performance Training Asymmetry for Clean and Noisy Speech

Returning to the summary data (Figures I.1 and II.2), consider the effects common to the performance of the two frontends, and in particular, the interaction between the addition of noise and the train/test combination. The inclusion of noise causes a major reduction in recognition performance in all conditions, but there are consistent differences depending on whether the system is trained on clean speech or noisy speech. When the system is trained on noisy speech and tested on noisy speech (large, filled squares and circles), the average decrement is around ten and eight percent for the DFT and SAS systems, respectively. When the model is trained on noisy speech and tested on clean speech (small, open circles and triangles), the decrement is never larger than for the system trained and tested on noise, and the decrement decreases substantially as codebook size increases. This indicates that the system is learning useful properties of the speech when it is presented in noise, and not simply characteristics of signal and noise in combination.

In contrast, when the system is trained on clean speech and tested on noisy speech (small, filled circles and triangles), the system suffers a further decrement of about five percent and it only recovers slowly as codebook size increases. This indicates that the characteristics of the signal learned from clean speech alone do not generalize well. The interaction suggests that, when a system is intended for use in noisy as well as quiet environments, the average performance might be improved by the simple expedient of adding noise to the training stimuli and including these noisy copies in the training set as if they were independent tokens. Performance on clean speech will undoubtedly decline a little but this decrement may be more than offset by improved performance on noisy speech. In effect, the inclusion of the noisy speech in the training set causes the system to focus on characteristics that are more appropriate for the eventual task.

If the SAS frontend assists in focusing the system on noise resistant characteristics of the speech, then including noisy speech

samples in the training set may lead to discrimination performance in which the SAS shows a greater advantage over the DFT.

## C. Performance and Time-Window Size

In the simplest HMM learning paradigm, the time window is an individual frame of the spectrogram and the size of the codebook vector is the same as the size of the spectrogram vector (i.e. 16). Recently, Iwamida et al (1989) found that a wider time window (seven spectrogram frames) led to better discrimination on a variety of phoneme recognition tasks, when lvq was used to generate the HMM codebook. The seven frame window produces larger reference vectors (16 by 7) and it was argued that this provided the vector quantizer with better information about the signal. There was some question as to whether the larger vectors would show an advantage with noisy speech, and whether the signal information would have the right form for generalizing from clean speech to noisy speech and vice verse. Consequently, we ran the main learning and test conditions using both window sizes.

The data for the small window size are presented in Figures II.4 and II.5 for the DFT and SAS systems, respectively. A comparison of the two systems comparable to that in Figure II.3 is shown in Figure II.6. When the DFT system is trained and tested on clean speech there is a very small advantage for the one-frame window. However, in general, the seven-frame window leads to much better performance. In particular, the seven-window system learns the speech better when it is presented in noise, and the learning transfers when the system is required to recognize clean speech. Similarly, having learned from clean speech it performs better than the one-frame system on noisy speech. The one exception is the case where a small codebook is used for training on clean speech and testing with noisy speech. In this case performance is very poor for both window sizes.

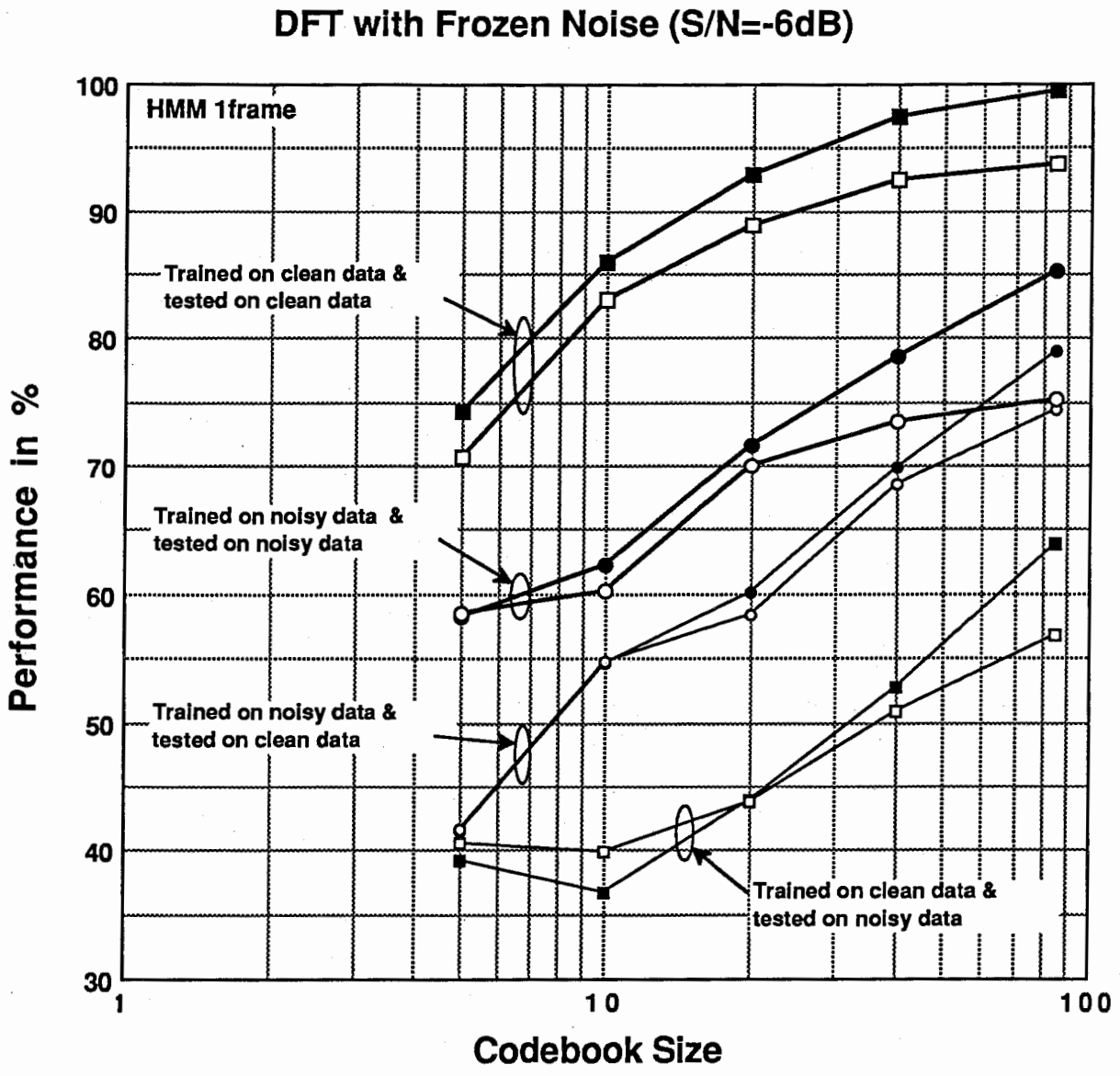


Figure II.4 Results for the DFT. Open symbols and filled symbols indicate results for the open and the closed recognition experiments. The reference vector size is 16-channel by 1 frame.

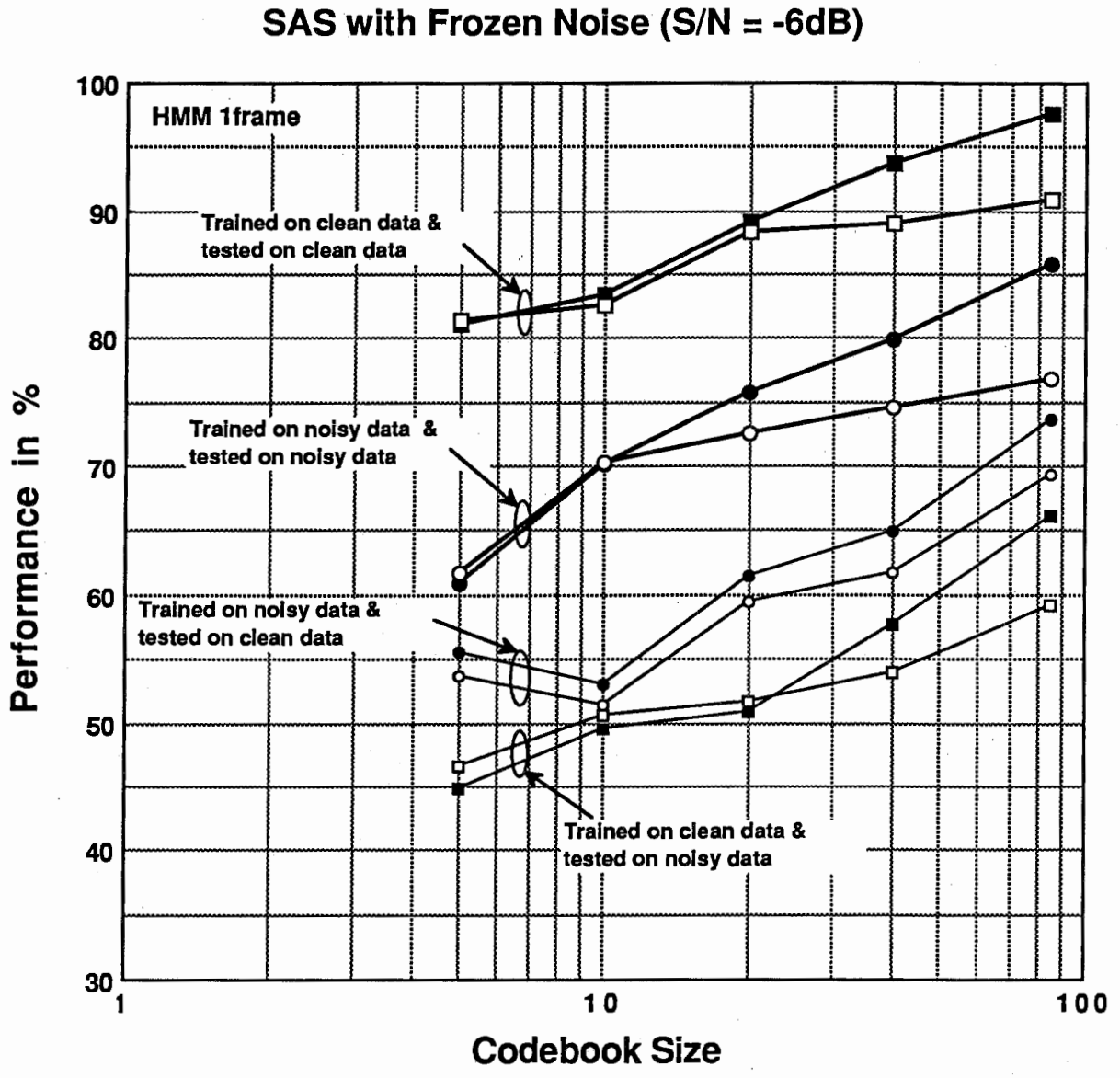


Figure II.5 Results for the SAS. Open symbols and filled symbols indicate results for the open and the closed recognition experiments. The reference vector size is 16-channel by 1 frame.

.

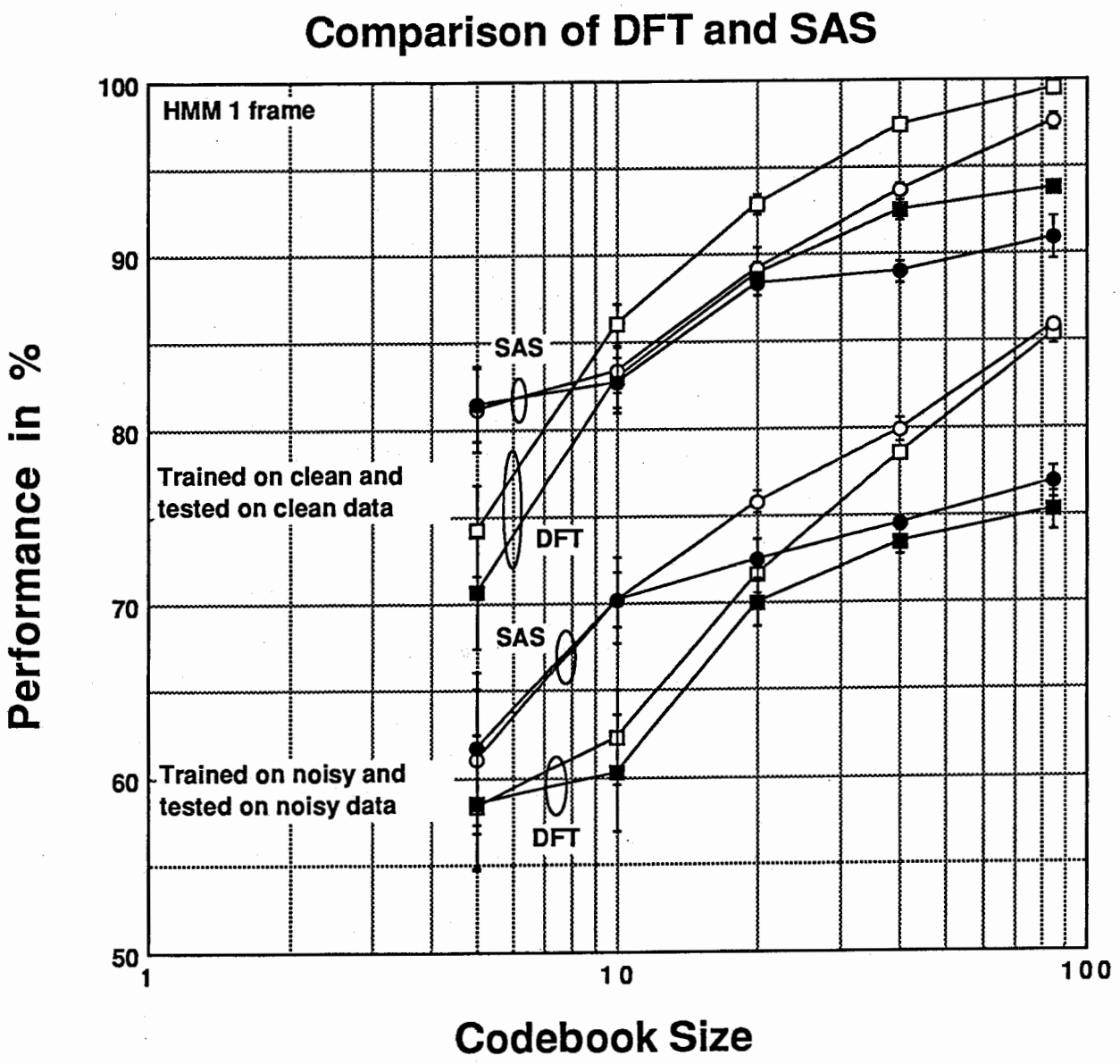


Figure II.6 A comparison of the DFT and SAS results for conditions where the system was trained and tested on clean speech(upper four curves) and for the conditions where the system was trained and tested on noisy speech (lower four curves). The reference vector size is 16-channel by 1 frame.

.

D. Performance in Noise

One of the surprising results was the performance of the DFT system when trained and tested on noisy stimuli. The results of Ghitza (1988) led us to expect that even a system with a large number of reference vectors would perform badly when the signal to noise ratio is small. In that study, recognition performance drops from over 90 percent in silence to around 50 percent when a background noise is introduced that produces an overall signal-to-noise ratio of 18 dB. At this point the advantage of the auditory frontend has increased from nothing to about 20 percentage points.

In the current study we began with clean speech and noisy speech with a range of signal-to-noise (S/N) ratios. Figure II.7 shows the spectrograms of five 'g' tokens (ga, ge, gi, go, gu) from the clean speech, with the DFT spectrograms in the upper part of the figure and the SAS spectrograms in the lower part of the figure. Both show the same general patterns but the formants are sharper in the SAS spectrograms. Figures II.8, II.9 and II.10 show spectrograms for the same tokens when the overall S/N ratio is 6, 0 and -6 dB. In Figure II.8 (S/N 6 dB), the speech features are attenuated but still distinguishable. In Figure II.9 (S/N 0 dB), the features are discernable in the SAS spectrograms but rather blurred in the DFT spectrograms. In Figure 10 (S/N -6 dB), the features are difficult to discern in all of the spectrograms. Following this and similar comparisons for /b/ and /d/ spectrograms, we chose to begin with noisy speech whose S/N ratio was 0 dB, as this appeared likely to produce the largest difference between the performance of the DFT and SAS frontends.

An analysis of the energy levels in the pure speech tokens is provided in Table II.3. It shows the mean levels and the standard deviations of the /b/, /d/, and /g/ tokens separately for both the training token set and the test token set. In each case, the first column presents the total power, and the second and third columns present the power of the consonant section before vowel

onset and the vowel section after vowel onset, respectively. The table shows that the average energy of the tokens is well matched.

The performance of the HMM recognizer with a codebook size of 40 was about 95 percent for clean speech and 85 percent for noisy speech when the S/N ratio was 0 dB, and it did not depend on the type of frontend. In order to reduce the performance in noise and provide more range in which to observe differences between the two types of frontend, we increased the level of the noise so that the overall signal to noise ratio was - 6 dB! This the level of noise in the conditions shown in Figures II.1 - II.6.

There are three important differences between this study and that of Ghitza (1988): Firstly, he used a different recognition system which does not appear to perform as well in noise as one might have expected. However, this is not an important difference for the current study. More important is the fact that the phoneme set is very restricted in the current study; the recognizer only has to make a distinction between three categories, 'b', 'd' and 'g'. When the phoneme set size is increased, there might well be a dramatic drop in performance in noise and an increase in the difference between performance based on the DFT and SAS frontends.

The third difference between the current study and Ghitza's is that, for convenience, we used only one noise sample and added this very same noise sample to each and every token in both the training and test sets. In retrospect, it would have been better to use a fresh noise sample for every speech token. The lack of variability in the noise from token to token is probably learned by the recognition system. This is especially unfortunate when the noise level is high because, in this case, the noise dominates the high frequency channels and the lack of change from token to token is particularly obvious. The result is that the recognition system can ignore a large portion of the spectrogram and concentrate on a few low-frequency channels. This would not be

an effective strategy with a large phoneme set but with a small phoneme set there would appear to be sufficient consistent information in this region to make the BDG discrimination.

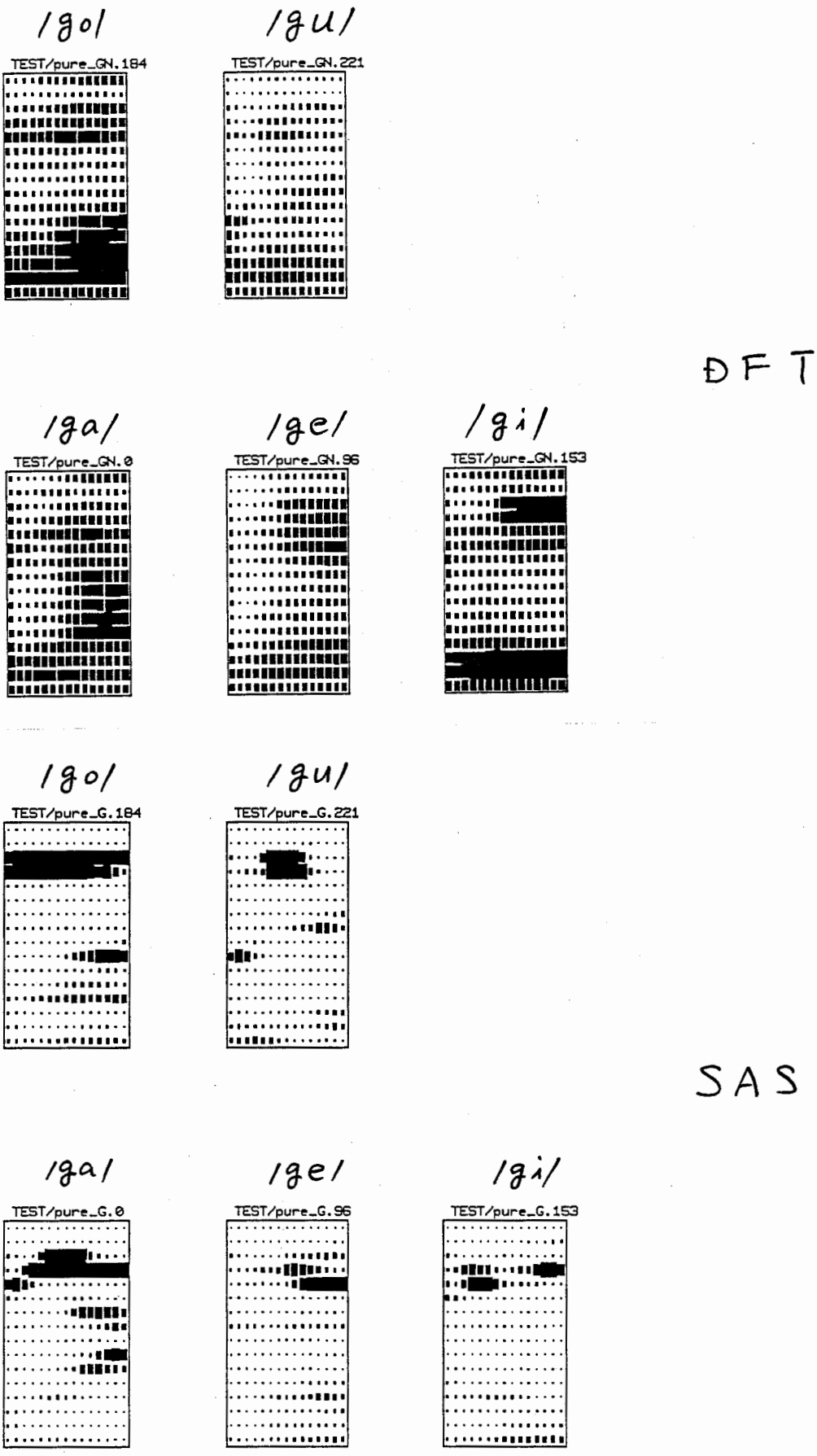


Figure II.7 16 by 15 spectrograms of five "g" tokens (/ga/ ,/ge/ ,/gi/ ,/go/ ,/gu) from the clean speech.

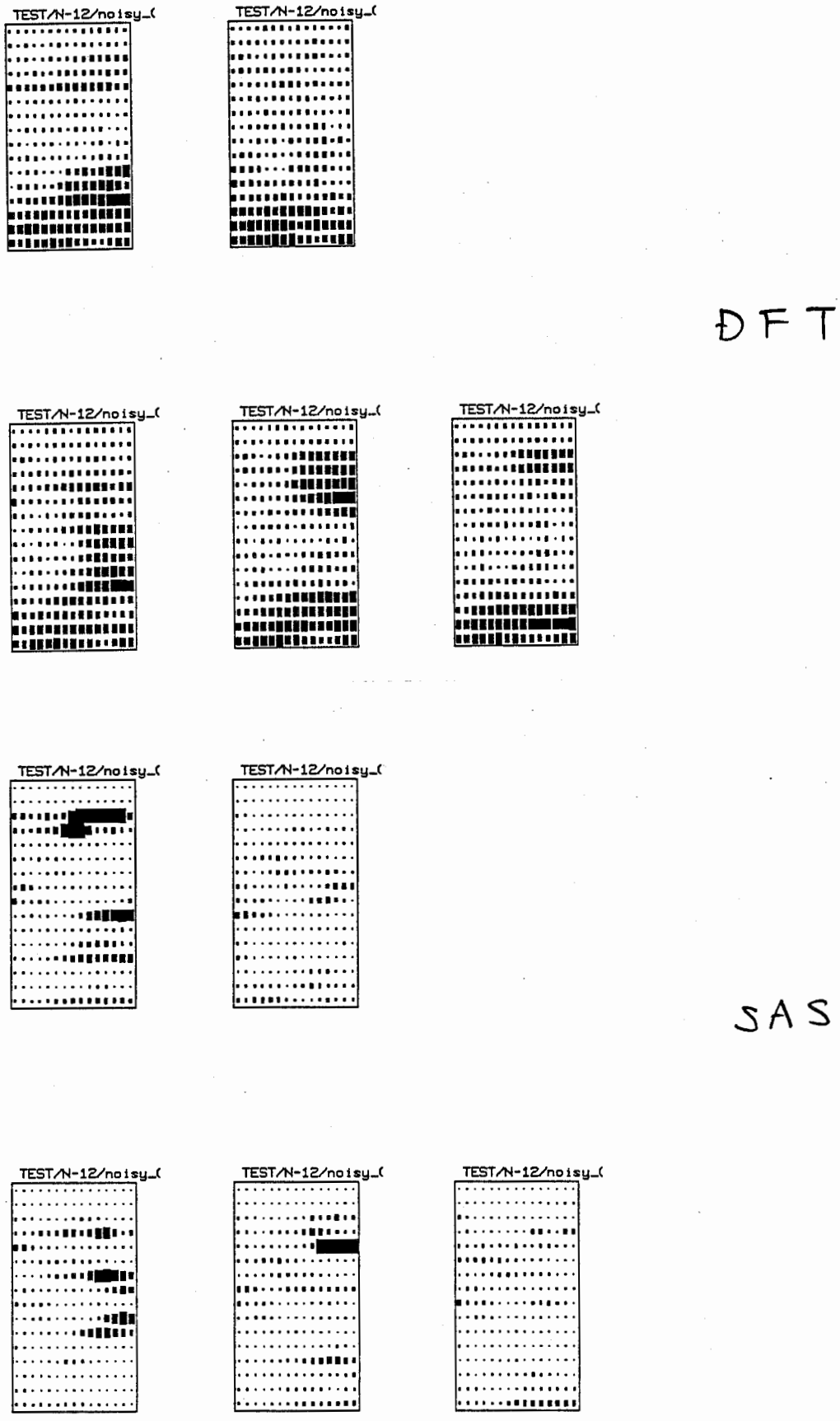


Figure II.8 16 by 15 spectrograms of five "g" tokens (/ga/ ,/ge/ ,/gi/ ,/go/ ,/gu) from the noisy speech where S/N = +6 dB.

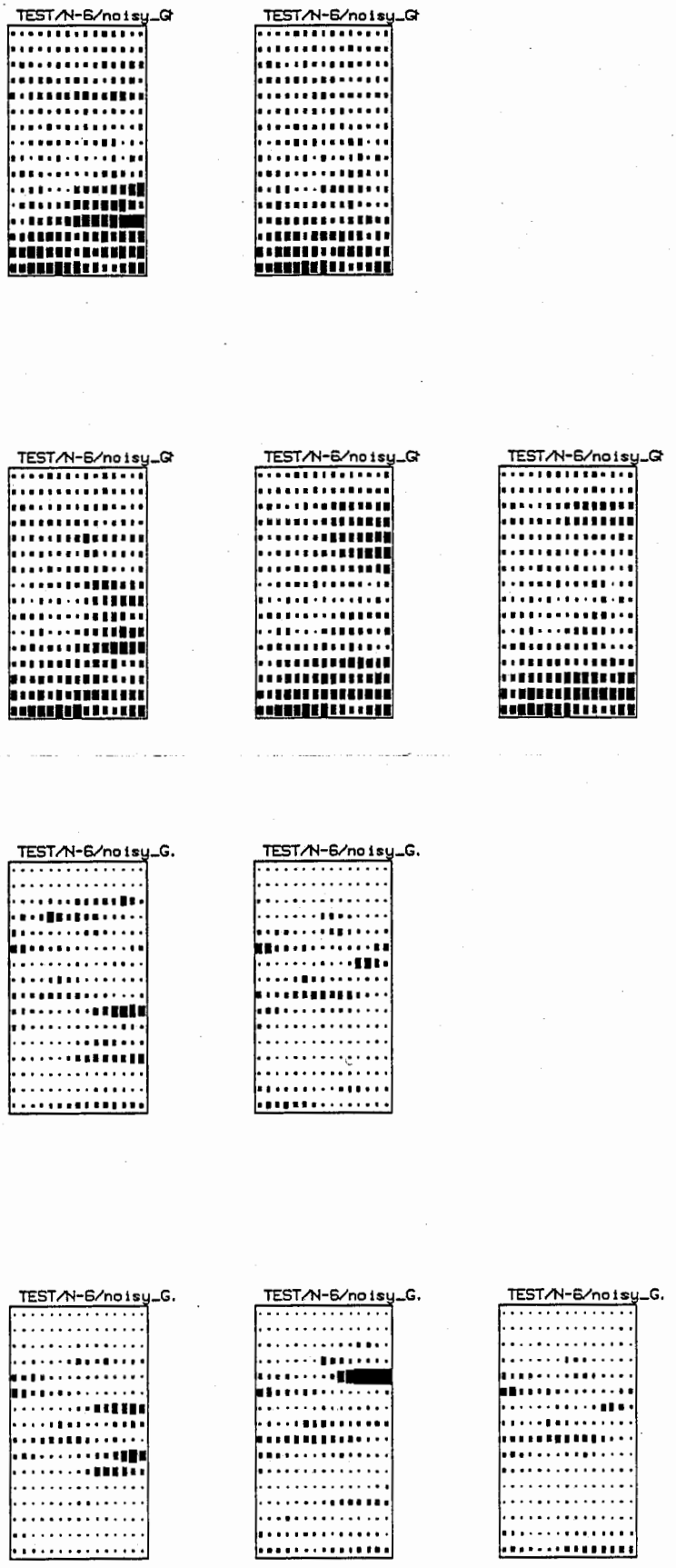


Figure II.9 16 by 15 spectrograms of five "g" tokens (/ga/ ,/ge/ ,/gi/ ,/go/ ,/gu) from the noisy speech where S/N = 0dB.

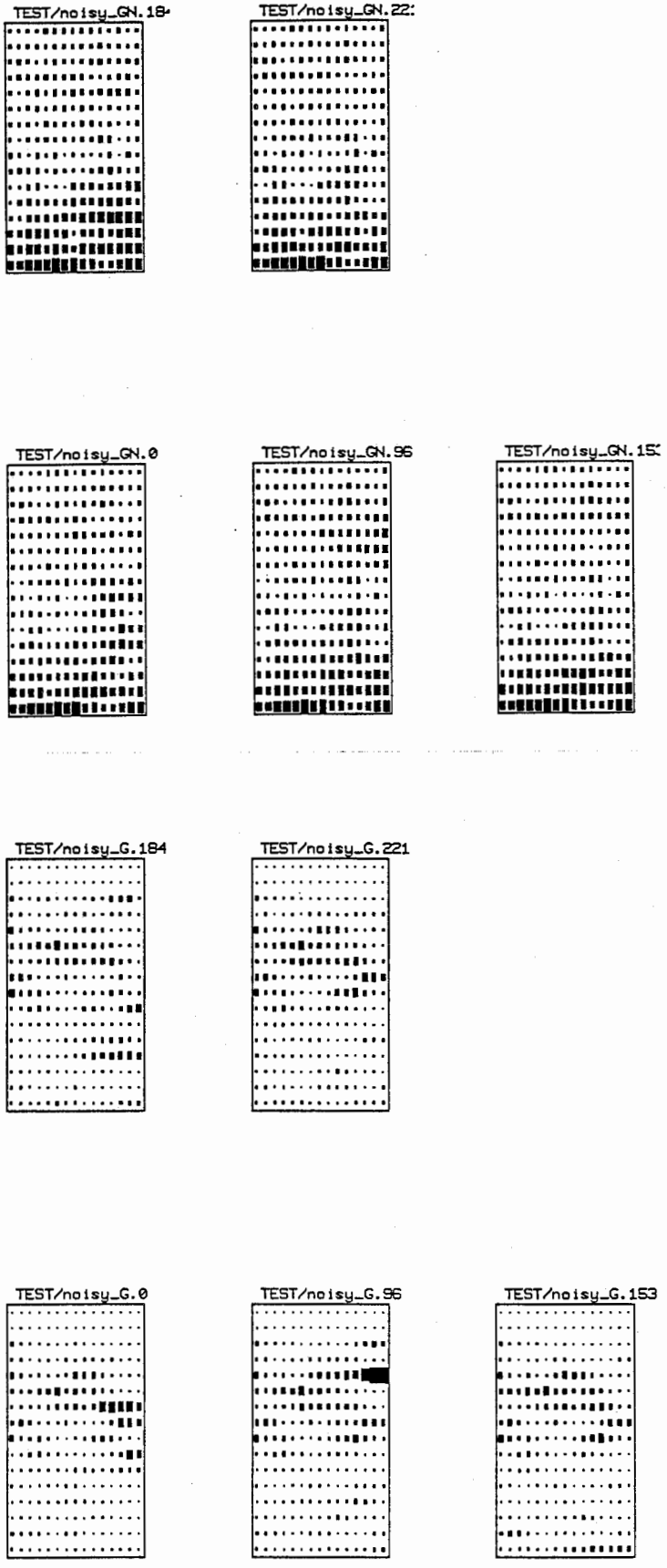


Figure II.10 16 by 15 spectrograms of five "g" tokens (/ga/, /ge/, /gi/ , /go/ , /gu) from the noisy speech where S/N = -6dB.

| | | Train_B | |
|---|---|---|---|
| | Total Power | Consonant part | Vowel part |
| Average | 61.50 | 51.50 | 64.10 |
| SD | 3.51 | 7.01 | 3.65 |
| Minimum | 53.80 | 13.90 | 56.70 |
| Maximum | 70.10 | 63.40 | 73.10 |
| | | | |
| | | Train_D | |
| | Total Power | Consonant part | Vowel part |
| Average | 62.30 | 50.80 | 65.00 |
| SD | 3.21 | 6.89 | 3.29 |
| Minimum | 54.30 | 31.10 | 72.40 |
| Maximum | 69.40 | 63.80 | 56.90 |
| | | | |
| | | Train_G | |
| | Total Power | Consonant part | Vowel part |
| Average | 62.50 | 55.00 | 64.40 |
| SD | 2.87 | 8.94 | 3.12 |
| Minimum | 54.50 | 17.10 | 56.40 |
| Maximum | 68.50 | 66.30 | 70.70 |
| | | | |
| | | Test_B | |
| | Total Power | Consonant part | Vowel part |
| Average | 61.60 | 51.10 | 64.10 |
| SD | 3.39 | 6.50 | 3.53 |
| Minimum | 53.90 | 21.60 | 56.80 |
| Maximum | 70.50 | 65.70 | 73.30 |
| | | | |
| | | Test_D | |
| | Total Power | Consonant part | Vowel part |
| Average | 63.00 | 50.50 | 65.70 |
| SD | 2.71 | 7.86 | 2.71 |
| Minimum | 55.00 | 11.70 | 57.30 |
| Maximum | 69.20 | 64.20 | 72.10 |
| | | | |
| | | Test_G | |
| | Total Power | Consonant part | Vowel part |
| Average | 62.50 | 54.80 | 64.50 |
| SD | 3.13 | 8.42 | 3.35 |
| Minimum | 54.20 | 14.20 | 56.60 |
| Maximum | 69.10 | 66.10 | 71.50 |
| | | | |
| | | | |
| | Total Power | Consonant part | Vowel part |
| PN-0 | 70.00 | 69.60 | 70.30 |
| PN-6 | 64.20 | 63.60 | 64.80 |
| PN-12 | 59.20 | 57.50 | 60.40 |
| PN-18 | 55.60 | 51.50 | 57.70 |

Table II.1 An analysys of the energy levels in the clean speech tokens and pink noises. Total power means the averaged power for whole token.(160msec). Consonant part and Vowel part means the averaged power of the first half and the last half part of each tokens.(80msec. each)

## CONCLUSIONS

1. The recognition tests should be repeated for the bdg task adding an independent pink noise sample to each speech token. The overall S/N ratio should be either 0 or 6 dB. This would be a more realistic test and it should improve the chances of observing a larger difference between the DFT and SAS systems.

2. The recognition tests should be repeated for a larger phoneme set to make the tests more realistic. It is also likely to increase the advantage of the SAS frontend. In this case, the most appropriate S/N ratio would appear to be 6 or 12 dB to start.

3. The recognition system should provided with a training set that includes both clean and noisy speech in an effort to improve the average performance of the system.

4. There are substantial differences in performance of the recognizer on the different phonemes, and interactions between phoneme and codebook size. The phoneme /d/ leads to the best performance and this performance is achieved even with small codebooks. The phoneme /g/ is recognized less well and performance increases slowly with codebook size. These differences might provide useful information when deciding how to retune the SAS model. It seems likely that there is currently a little too much suppression and that it should be reduced both in time and frequency.

5. With regard to the specific HMM recognizer used in this study, it is clear that the seven frame reference vector system is superior to the one frame reference vector system. However, this difference may be reduced if the system is required to work with a larger phoneme set, and so the comparison should be repeated if a larger phoneme set is introduced.

## Acknoledgements

Much of this work was performed in September and October of 1989 while the first author was a visiting researcher at ATR. The Auditory Sensation Processor was developed at APU by Roy Patterson and John Holdsworth. The software itself was written by John Holdsworth with the assistance of Paul Manson.

The authors wish to thank to Hitoshi Iwamida for providing us the HMM program. They are grateful to Erik McDermott for his fruitful discussions.

## References

[1] Carney,L.H. and.Yin,C.T. (1988) "Temporal coding of resonances by low-frequency auditory nerve fibers: Single fiber responses and a population model," Journal of Neurophysiology 60, pp.1653-1677

[2] Ghitza,O. (1988), "Temporal non-place information in the auditory-nerve firing patterns as a front-end for speech recognition in a noisy environment," Journal of Phonetics, vol. 16, No. 1, pp.109-123

[3] Holdsworth,J. (1989) "Two-Dimentional Adaptive Thresholding, " Applied Psychology Unit Technical Report.

[4] Iwamida,H., Katagiri,S., McDermott,E., and Tohkura,Y.," A Hybrid Speech Recognition System using HMMs with and LVQ-trained Codebook", ATR Technical Report TR-A-0061, pp.1-18

[5] Moore,B.J.C. and Glasberg,B. (1983), "Suggested formulae for calculating auditory filter bandwidths and excitation patterns," J.Acoust.Soc.Am. vol. 74, pp.750-753

[6] Patterson,R.D.,Nimmo-Smith,I.,Holdsworth,J., and Rice,P. (1988), "Spiral Vos Final Report, Part A: The Auditory Filterbank," Cambridge Electronic Design, Contract Report (APU 2341)

[7] Waibel,A., Hanazawa,T., Hinton,G., Shikano,K., and Lang,K., (1988) "Phoneme Recognition:Neural Networks vs. Hidden Markov Models," Proc. ICASSP'88, pp.107-110

[8] Wilpon,J.G. and Rabiner,L.R.(1985), " A modified K-means clustering algorithm for use in isolated word recognition," IEEE Trans. ASSP-33, pp.587-594

## Appendix 1 Speech Data & Noise Data

Speech data used in the experiments are CV-syllables extracted from a large database of 5,240 common Japanese words, which were uttered in isolation by a native male Japanese speaker (MAU). All utterances were recorded in a soundproof room and digitized at a 12kHz sampling rate. The database were then split into a training set and a test set of 2620 utterances each, from which all CV-syllables including /b/, /d/ or /g/ were extracted using manually selected acoustic-phonetic labels provided with the database.

Table A-1 shows the number of tokens for each /b/, /d/ and /g/ in a training set and in a test set.

| Token | Training set | Test set |
|---|---|---|
| ba | 53 | 66 |
| be | 21 | 21 |
| bi | 39 | 34 |
| bo | 43 | 43 |
| bu | 62 | 63 |
| **/b/** | **218** | **227** |
| da | 89 | 81 |
| de | 29 | 27 |
| do | 84 | 71 |
| **/d/** | **202** | **179** |
| ga | 96 | 96 |
| gi | 57 | 53 |
| gu | 31 | 31 |
| ge | 37 | 41 |
| go | 39 | 31 |
| **/g/** | **260** | **252** |

Table A1-1 Number of tokens.

Noise data used in the experiments were the pink noise (20Hz to 20kHz) generated by a signal generator (B&K 1049). The pink noise were sampled at 12 kHz sampling rate.

## Appendix 2 Spectrum Analysis

### 1. DFT spectrogram

Input speech was hamming windowed (21.5 msec.) and a 256-point FFT computed DFT power spectrum every 5 msec. Then, 16 mel-scale coefficients were computed from the power spectrum and adjacent coefficients in time collapsed resulting in an overall 10 msec. frame rate. Normalization has NOT been performed.

### 2. SAS spectrogram

Input speech was analyzed with 32-channel gammatone filter (200Hz to 4kHz) and the SAI (Stabilized Auditory Image) was computed every 1.25 msec. Then, SAS spectrogram (32 by 128) were computed and reduced to 16 by 16 vector by averaging adjacent pair of channel and averaging ad jacent sets of eight frames.

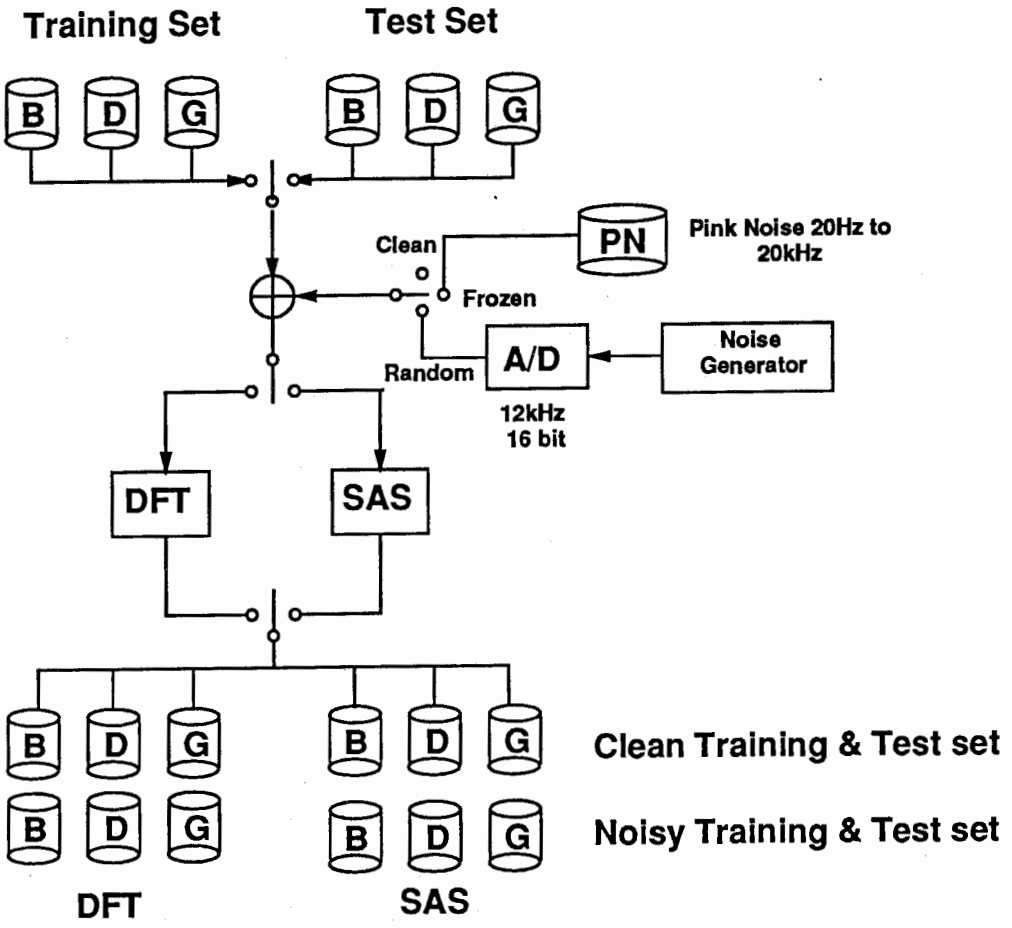


Figure A2-1 Block diagram of the DFT and SAS analysis

## Appendix 3 Hidden Markof Model

In Figure A3-1, the block diagram of phoneme recognition system using HMM is shown.

K-means clustering procedure was used to make a codebook. The input vectors for the clustering procedure were either a 16 channel by 7 frame partial vector or a 16 channel by 1 frame partial vector. When seven frame vector was used, nine partial vectors were obtained from one token (16 channel by 15 frame).

An HMM with four states and six transitions was used in this study.(Figure A3-2) The transition probabilities of the HMMs ($a_{ij}$) are all initialized to have equal values. The initial values $b_{ik}$ are set , for each code k, at the number of observations of the code k, diviated by the number of observations of all codes. The Baum-Welch algorithm, based upon maximum likelihood estimation, is used to train the HMMs. The number of iterations were set at 7. Floor value were set on the output probabilities at $10^{-6}$ to avoid errors caused by zero probabilities.

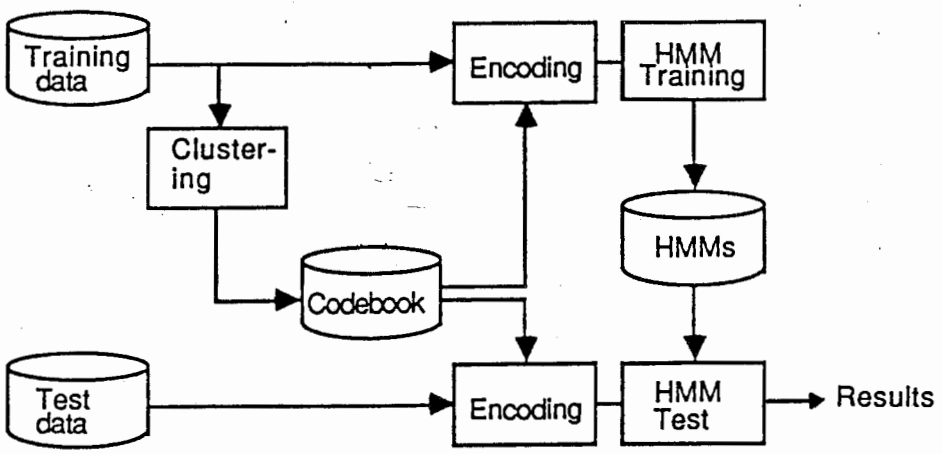


Figure A3-1 Block diagram of phoneme recognition system using HMM.

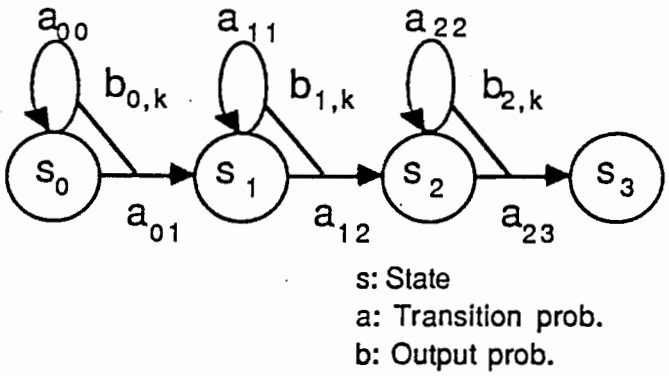


Figure A3-2 Phoneme model structure.