Abstract—The quantization distortion of vector quantization (VQ) is a key element that affects the performance of a discrete hidden Markov modeling (DHMM) system. Many researchers have realized this problem and tried to use integrated feature or multiple codebook in their systems to offset the disadvantage of the conventional VQ. However the computational complexity of those systems is then increased.

Investigations have shown that the speech signal space consists of finite clusters that represent phoneme data sets from male and female speakers and reveal Gaussian distributions. In this paper we propose an alternative VQ method in which the phoneme is treated as a cluster in the speech space and a Gaussian model is estimated for each phoneme. A Gaussian mixture model (GMM) is generated by the expectation-maximization (EM) algorithm for the whole speech space and used as a codebook in which each code word is a Gaussian model and represents a certain cluster. An input utterance would be classified as a certain phoneme or a set of phonemes only when the phoneme or phonemes gave highest likelihood. A typical discrete HMM system was used for both phoneme and isolated word recognition. The results show that the phoneme-based Gaussian modeling vector quantization classifies the speech space more effectively and significant improvements in the performance of the DHMM system have been achieved.

I. INTRODUCTION

Hidden Markov modeling (HMM) is commonly used for speech recognition. A discrete HMM speaker-independent isolated word recognition system can be described as a two-step modeling process. The first step, vector quantization (VQ), is used to classify the speech signal space into \( N \) regions, where \( N \) is the codebook size or number of models generated in this step. Each region is represented by a typical vector, usually the centroid vector for that region. The codebook is then composed of these \( N \) typical vectors. The second step—HMM—is used to produce a set of reference models that represent the possible sequences of the quantized observation vectors from the codebook generated by the first step. The discrete HMM recognizer is commonly used not only because of its reasonable performance and potential low computation cost, but also because it is a parametric model of the speech signal that can model various events (phonemes, syllables, etc.) in the speech signal [1].

The most common vector quantization algorithm is the Linde–Buzo–Gray (LBG) algorithm proposed by Linde et al., in 1980 [2]. Although the LBG algorithm has the advantages of being simple in concept and implementation with low computational costs, it is only locally optimal with relatively high distortion error. The latter is a consequence of the rigid geometric partitioning of the space, which may not be the most appropriate. Another disadvantage of the LBG algorithm is that the classification is not concerned with the statistical features of the speech signals, whereas the HMM is. In other words, the classification performed by the LBG algorithm depends on a geometric distance measure rather than a statistical measure and is therefore of limited applicability to the HMM, which produces a model based on the statistical features of the training sequence. Inappropriate vector quantization of the training data leads to the loss of some statistical information from the signals and eventually decreases the recognition accuracy of the whole system.

The majority of researchers have realized that an inappropriate vector quantization affects the system performance for a discrete HMM speech recognizer, and proposed a number of alternative strategies, such as the use of multiple speech features, multiple codebooks, etc., for the first step modeling [3]–[7] and the continuous [1] and semicontinuous HMM [7] as a means of incorporating this in the second-step modeling. Huang’s semicontinuous hidden Markov model (SCHMM) [7] is the most successful of these. Most of the strategies [4]–[6] try to decrease the quantization distortion to improve the system performance. The reason the SCHMM works well is that it combines the VQ codebook construction with the HMM training into a single algorithm and the VQ code book is optimized with respect to the HMM likelihood maximization. In other words, the stochastic information of the signal is included in the VQ codebook construction.

Including statistical information about the speech signal in the first step modeling is an effective way to improve the performance of recognizers. It allows us to classify the speech signal space dependent on its statistical distribution and, also, match the first step modeling to the HMM, which is a statistical modeling technique. Previous experiments [8], [9] gave us improved system performance in a speaker-independent isolated word recognition system when we used the expectation-maximization (EM) algorithm [10] instead of the conventional vector quantization method, the LBG algorithm, as used by Rabiner et al. [3].

The EM algorithm may be used for obtaining the maximum likelihood parameters for a data set modeled by a mixture of...
probability density functions. In this paper, we use Gaussian mixture models for the speech data. Considering each model in the mixture as a cluster of the data space, we in fact use the EM algorithm to classify the signal space into clusters of which each cluster is represented by a model in the mixture. In other words, the EM algorithm performs a vector quantization and generates a codebook in which each code word is a Gaussian model composed of a mean vector, a covariance matrix, and a mixture weight. It is obvious in such a vector quantizer that the EM classifier decodes the speech feature vectors in a stochastic sense, and carries this stochastic information into the HMM. Further investigation has shown that the multidimensional distribution of phonemes reveals a Gaussian-like hyperellipsoidal structure. It implies that the vector quantization of speech features could be performed based on each phoneme being treated as a region or cluster. In other words, we can generate a codebook in which each code word represents a phoneme cluster. In this paper we describe a speech recognizer in which a phoneme based vector quantization using Gaussian mixture modeling was performed and a discrete HMM (DHMM) was employed to generate the reference models. Comparison of the performance between the conventional discrete LBG-HMM, GMM-HMM, and phoneme based VQ-HMM has shown that the last method yields the best results.

In this paper, we will first describe an investigation of the phoneme space and conclude that the speech signal space can be modeled as consisting of phonetic clusters. The pictures will illustrate our observations of the multidimensional phoneme space and demonstrate that the phoneme-based Gaussian mixture modeling is a suitable approach for the first step modeling. Then we will introduce the use of the EM algorithm as the alternative to the LBG algorithm and give some practical details. Next, we will describe our experimental design, which includes the database sets used in our experiments. Following this, our experimental results will be given for phoneme classification and isolated word recognition comparing the different methods. Finally, we present a discussion of the results and conclusions.

II. PHONEME SPACE

To visualize the 12-dimensional (12-D) space, we project the points to a plane that is then mapped to the computer screen giving us a view of the space. To give us a better understanding we normally consider more than one view, although only one representative view is shown for the figures.

Phonemes were manually segmented from the speech waveform. The vowel sounds were easiest to extract with the plosives or stops being the most difficult due to their short duration, sometimes shorter than one analysis frame. It is conceivable that the plosive and stop data have been affected by the surrounding aspirate or closure. In order to extract the short consonants and obtain their features, we segmented the short stops with some visible surrounding waveform. This has minimal effect on the parameter estimation of short consonants because the stops and plosives display as clusters separate from the adjacent aspires or closures when the features are visualized.

The visualization process is as follows. A set of 12-D fast Fourier transform (FFT) filter bank features were produced for each phoneme. We projected the features to a two-dimensional (2-D) computer screen together with the ellipsoid which represents the Gaussian model estimated from the data class. The covariance matrix of the Gaussian model was used to generate the eigenvalues and eigenvector which describe the shape of the hyper-ellipsoid in the space. To illustrate the Gaussian distribution derived from each phoneme class, the points satisfying the following formula are plotted as the hyperellipsoid

$$x^TC^{-1}x = 1$$

where $x$ is a multidimensional point and $C$ is the covariance matrix of Gaussian model. For clarity, we plot the locus of points describing the ellipses formed by two of the 12 eigenvectors as the axes. This will result in 66 ellipses for all possible eigenvector pair combinations of 12. We plot only 15 ellipses formed by the eigenvectors corresponding to the six largest eigenvalues. The phoneme label notation used throughout this paper is the ASCII CMU phonetic label.

If the cluster of segments of speech corresponding to a phoneme extracted manually from an utterance is projected from 12 dimensions to the computer screen, it may be seen that the phoneme clusters are plausibly described by Gaussian distributions. Since this holds for all the projections tried, we feel justified in modeling the 12-D data by a 12-D Gaussian model. Fig. 1 shows the data of two vowels, /AA/ and /IY/, as well as their Gaussian models. It was also observed that consonant sounds are visually more separable than vowels. This explains why the recognition accuracy of consonants is normally higher than that of vowels. The diphthong data distribution displays as two partially overlapped Gaussian clusters and trajectory plots of the diphthong sound always travel from one cluster to the other. Observations for the different classes of consonants, say, stops, fricatives, and nasals, revealed that they occupy separate regions in the speech signal space. It was also observed that the three main classes of phonemes—vowel, consonant, and the semivowels—lie in different regions of the space and are distinct. This point is also illustrated by the fact that there was little confusion of phonemes between different classes in the phoneme recognition experiments. Another important observation is that the same phoneme data from speakers of different sex occupy partially overlapped regions highlighting the spectral transformation from one class of speakers to another. In short, a vowel or a consonant can be described by a single Gaussian model, while a diphthong is best described by two partially overlapped Gaussian models.

Based on the above observations, we claim that the speech signal space is a finite cluster space and we propose that each phoneme data class can be modeled as a Gaussian cluster for the first step, VQ modeling.

III. ALTERNATIVE VECTOR QUANTIZATION STRATEGY

The drawbacks of conventional DHMM strategy have caused many researchers to seek alternative strategies to improve the system performance.
If we consider the way in which speech is produced, it is reasonable to hypothesize that different utterances of the same speech sound will form a cluster around some center. The variations about the mean will occur at random when a large population of speakers is considered, so the points in the cluster will be distributed according to a multidimensional Gaussian probability density function. We have seen the rationalization for this hypothesis in the second section and observed that the cluster utterances represent phoneme subwords.

This view of the speech-production process suggests that VQ of the speech space is better performed on the basis of a Gaussian mixture model (GMM), in which the data is clustered around a mean value according to a Gaussian distribution, and each cluster is assigned a weight corresponding to the relative frequency of data belonging to the cluster.

A Gaussian mixture model is defined as

\[ f(x) = \sum_{i=1}^{N} p_i g(x; \mu_i, \Sigma_i) \]

where \( g(x; \mu_i, \Sigma_i) \) is the Gaussian probability density function with mean \( \mu_i \) and covariance matrix \( \Sigma_i = \left( \sigma_{ij}^k \right) \), \( x \) is a random \( D \)-dimensional vector, \( x = (x_1, x_2, \ldots, x_D) \), and the \( p_i \) are weights that describe the relative likelihood of classes being generated from each of the clusters and must satisfy \( \sum_{i=1}^{N} p_i = 1 \), where \( N \) is the number of classes.

In cases where the distance between means is large in comparison to the square roots of the variances, this model describes a set of isolated clusters of ellipsoidal shape. As the distances between means decreases, the isolation of the clusters is progressively reduced until they merge.

The GMM can also be given an alternative interpretation as a set of local distance measures on the speech space. Each distance measure is applicable to a region surrounding the mean of one of the Gaussian clusters, in which the Mahalanobis distance derived from the covariance matrix of the Gaussian distribution is used.

In order to estimate the parameters of the GMM, the EM algorithm [10], [11] was employed. The EM algorithm is an iterative process where each iteration consists of an expectation (E) step followed by a maximization (M) step. The EM algorithm partitions the signal space based on the probability distribution of the space, and assumes that observations originate from Gaussian sources. Each cluster in the Gaussian mixture model then forms a quantization code book entry.

The initialization of GMM’s in the EM algorithm is important. The weights \( p_i \) are simply initialized by \( 1/N \), where \( N \) is the number of data classes. The covariance matrices are initialized to the unit identity matrix. This implies the initial Gaussian model is a unit hypersphere. The initialization of mean vectors can either be performed by random allocation of input vectors or using alternative methods, like K-means clustering [2] or the move-means algorithm [11].

**IV. EXPERIMENTAL DESIGN**

Two English databases were used for the training and testing. The first database is the studio quality speaker-independent connected-digits corpus (TIDIGITS) from the National Institute of Standards and Technology (NIST) in the United States. The data is sampled at a 20-kHz sampling rate and digitized to 16-b resolution. A subset of the data base used in our experiments called DAT1 comprised a small vocabulary of eleven isolated digits (from zero to nine and oh) spoken by 112 speakers (55 males and 57 females) and test data spoken by 113 different speakers (56 males and 57 females). The second database is the TI46-word speaker-dependent isolated word corpus also from NIST. This data base called DAT2 comprises 46 isolated words, 10 digits, 10 computer command words, and 26 English alphabets. The data is sampled at 12 500 Hz and digitized to 14-b resolution. There are 16 speakers (8 male and 8 female) in the data base, and each word was repeated 26 times by the speakers. The first 10 repetitions were used as the training set and the remaining 16 as the testing set.

The FFT of the speech data was computed every 10 ms with a 25-ms Hamming window. The FFT coefficients were binned into 12 Mel-spaced values to produce 12-D feature vectors corresponding to the frequency range from 60–5000 Hz. Our experiments included phoneme recognition and two types of isolated word recognition, speaker-independent isolated digit recognition using DAT1 and speaker-dependent isolated alphabet recognition using DAT2.

According to our observation that the phoneme data from male and female speakers occupy different regions, the training sequence was divided into two parts based on the sex of the speaker. The phonemes were manually extracted from the training sequence. There are 19 phonemes from DAT1 and 24 from DAT2. Table I is the list of phonemes from the two data bases. The reader should note that, strictly speaking, the sounds /D-ZH/ and /T-SH/ are not single phonemes, but from...
A. Phoneme Recognition

Our improved DHMM system was first used to recognize the phonemes that were extracted manually from the DAT1 and DAT2 data sets. The Gaussian models obtained individually from the phonemes were used to initialize the EM algorithm. The data for all phonemes was used as the training sequence. The EM algorithm iterations were run with the E step followed by the M step to recalculate the mixture weights only, keeping the means and covariances fixed. After a Gaussian mixture model was generated by the EM algorithm, it was used as a codebook to quantize the observation sequence. The codebook includes 44 code words for DAT1 and 58 code words for DAT2. The output of this VQ stage was followed by DHMM section to generate two HMM’s for each phoneme, one for male speakers and another for female speakers. The input string is classified as the phoneme whose model gives the maximum likelihood. A three-state constrained left-to-right sequential HMM structure was employed for this task.

Phoneme classification can also be performed by using only VQ without HMM [12]. In a VQ-only phoneme recognizer, the codebook is composed of the Gaussian models as described above. A likelihood calculation for each frame of data is averaged across all trained phoneme models. Recognition is then based on the model with maximum average likelihood. The problems with the VQ-only recognition is that there is inadequate modeling of context-dependencies as well as inadequate modeling of diphthong sequences. The HMM addresses this problem by modeling possible left and right contexts and the transitions in a diphthong utterance.

B. Isolated Word Recognition

Consider the $M$ Gaussian models from the individual phonemes of DAT1 (i.e., $M = 44$). In this case, it is reasonable that the code book should contain more vectors than $M$ because of the presence of background noise, silence between the speech utterances, as well as some detailed sound not included in the phoneme extraction. In other words, the phoneme space is not the whole speech space, it is a subset of the observation sequence. Thus, the EM algorithm was used to generate a Gaussian mixture model that has $N$ ($N > M$) Gaussian models for the whole training sequence. The previously obtained $M$ Gaussian models were used to initialize the EM training procedure. The means and covariance matrices of the $M$ Gaussian models were fixed in the estimation iterations but the probabilities (or weights) of the $M$ Gaussian models were recalculated by the EM algorithm. For the remaining $N-M$ Gaussian models the means, covariances and mixture weights were estimated by the EM algorithm. After EM training we had a Gaussian mixture model of $N$ Gaussian models. The construction of the codebook is shown in Fig. 2. In the figure, each circle represents a phoneme cluster and the letters in the circle indicate which phoneme it is and whether the data is from a male or female speaker. For instance, /AA/ indicates the phoneme data /AA/ from a female speaker. The lower box represents the $N$-element code book. The $G_1$–$G_N$ are the $N$ Gaussian models in the Gaussian mixture generated by the EM algorithm.

In the second modeling step, a five-state constrained left-to-right sequential HMM structure was used to derive the reference models, two for each word in the vocabulary, one for male speakers and one for female speakers. The Baum–Welch reestimation algorithm was used to train the models. In the recognition phase, the probability of generating the observed string is computed for each reference model using the Viterbi algorithm.

Three different VQ methods were used in our experiments. Method VQ1 is the conventional VQ in which the codebook was created by the LBG algorithm and the Euclidean distance is the measure. In method VQ2, a Gaussian mixture model as the codebook was generated by the EM algorithm and the log likelihood was the measure. The codebook size for both VQ1 and VQ2 is 64. The third method, VQ3, is the phoneme
Fig. 2. Construction of the codebook.

<table>
<thead>
<tr>
<th>Database</th>
<th>VQ1</th>
<th>VQ2</th>
<th>VQ3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAT1</td>
<td>59.3</td>
<td>63.5</td>
<td>71.8</td>
</tr>
<tr>
<td>DAT2</td>
<td>54.7</td>
<td>60.1</td>
<td>69.7</td>
</tr>
</tbody>
</table>

TABLE II
RECOGNITION ACCURACY RATES FOR PHONEME RECOGNITION

based VQ described in the previous section in which the 44 Gaussian models for DAT1 and 54 for DAT2 were generated individually and were combined to form the Gaussian mixture model codebook by the EM algorithm. The difference between VQ2 and VQ3 is that in VQ2, the GMM was generated from the whole training sequence directly while in VQ3 case, M Gaussian models of the GMM were specifically initialized and fixed during the EM training procedure. The details of VQ2 can be found in the literature [8], [9].

V. RESULTS AND DISCUSSIONS

Shown in Table II is the recognition accuracy rates of the phoneme classification. From the results we conclude that using the GMM as the codebook is more reasonable than the conventional codebook created by the LBG algorithm. Not surprisingly, the VQ3 classifies the phoneme space much more efficiently than VQ1 and VQ2.

We observed that in the recognition phase the diphthong sequence always transferred from one Gaussian model (or state) to the another, giving almost identical recognition accuracies as the vowels. The recognition accuracies, however, decreased by 15–20% when we used one Gaussian model to fit each of the diphthongs. These results illustrate that it is reasonable to generate two Gaussian models for each diphthong.

We observed that the utterances from the female speakers were never incorrectly recognized as the models generated from the male speaker’s data and vice versa. It suggests that the data from different speakers based on sex occupy separate regions and should be modeled separately.

Table III shows the results for isolated word recognition. In order to compare the results from the three different VQ methods, the codebook sizes are fixed at 64, 128, and 256, which match the requirement for a codebook size of $2^L$ when using the LBG algorithm. From the results, we see that VQ2 gave better performance than VQ1 with VQ3 giving the best recognition accuracies. This is consistent with our expectation that the Gaussian mixture model is a good description of speech features and that the phoneme based vector quantization using the GMM classifies the speech space more effectively.

The codebook size is important for the overall system performance. A small codebook will yield a low recognition accuracy because of the high vector quantization distortion, while a larger codebook leads to better system performance but at a higher computational cost. On the other hand, the codebook size is limited because the speech signal space is not infinitely divisible. An excessively large codebook leads to excessive classification of the signal space, so that several codebook vectors will correspond to the same region that describes the same part of an utterance, say, a phoneme. This will cause confusion for the finite-state HMM since the partitioning will be “overtrained” (in the same sense that an artificial neural network can be overtrained). The recognition accuracy then becomes dependent on sufficient training data to generalize the HMM models from the “overtrained” partitioning. This upper bound limitation on the codebook size varies for the different VQ methods and different vocabulary. Generally speaking, the bigger the vocabulary is, the larger the codebook is required, but the more efficient the speech space is classified, the smaller codebook size is demanded.

From Table III, we can see that when the codebook size is increased from 128 to 256, the recognition accuracies of VQ1 were improved, those of VQ2 increased slightly, while the cases of VQ3 were worse. This fact indicates that phoneme-based vector quantization classifies the speech space more efficiently and works with a smaller codebook size. The same phenomenon was also observed by Dai [16]. He concluded that increasing VQ resolution does not necessarily imply a high recognition rate and explained that a very high VQ resolution implies the requirement of a very close match between the features of a training utterance and those of a test utterance from the same word class. We can expand on Dai’s explanation for the speaker-independent speech recognition case, where a very fine classification for the features of the training speakers will certainly not match the testing speakers well.
These results were also compared with a continuous density HMM system (CHMM), a tied mixture or semicontinuous HMM system (SCHMM), and a multi-VQ multi-HMM system (M-VQ M-HMM). Here, our phoneme-based VQ DHMM system is abbreviated as PBDHMM. Table IV shows the accuracies of the four systems for both isolated digit and alphabet recognition. In the CHMM system, the same five-state left-to-right constraint HMM structure as the DHMM was employed, and a Gaussian model with full covariance matrix was generated for each state of the models. The M-VQ M-HMM system is as described in Zhang et al. [15]. The tied mixture CHMM system, which is equivalent to the semicontinuous HMM system, was based on the above CHMM structure with a tied mixture of 128 Gaussian models. Compared with these state-of-the-art strategies, the DHMM system with phoneme-based VQ obtained comparable if not better results, using a simpler HMM structure, and a more direct and flexible feature-space modeling strategy.

The recognition computation required by the PBDHMM system is the same as the standard DHMM system, since the Viterbi decoder stages of both the systems are identical. It is well known that the computation complexity of the DHMM system is much lower than the CHMM system. Our PBDHMM system also requires less recognition time than the SCHMM system. For each input vector in a PBDHMM system, the VQ stage determines the single best codeword for the Viterbi decoder. However, the VQ of the SCHMM typically chooses the L best code words (normally L = 3–5) as output. This increases the computational costs in the Viterbi decoder. The computation needed by our PBDHMM recognizer mainly depends on the formula

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} \left[ \delta_{t-1}(i) a_{ij} \right] b_j(O_t)$$

where

$$O_t = \arg \max_{1 \leq i \leq M} f_i(x_t)$$

whereas the SCHMM decoder is based on

$$\delta_{t+1}(j) = \max_{1 \leq i \leq N} \left[ \delta_{t-1}(i) a_{ij} \right] \sum_{\eta \in \Omega(x_t)} \left[ f_j(x_t) b_j(\eta) \right].$$

More important, our proposed scheme is conceptually different than the SCHMM in that the feature-space modeling has been separated from the state modeling of the hidden Markov model. This modularity in our design can have obvious advantages in optimising the feature-space modeling (using more sophisticated clustering algorithms) as a separate process to optimising the state or time-dependency modeling (currently based on the basic HMM framework although other techniques [17], [18] using nonstationary states and trajectory modeling are proving to be superior).

The results we obtained can be compared with other recent reports [5], [14], [15]. Bocchieri and Wilpon [13] employed 38-dimensional delta-delta cepstrum (DDCEP) feature vectors and obtained 98.6% recognition accuracy for the TI connected digits and 80.8% for the E-set letters. They also reported that when the feature vector sizes were increased from 12 to 32 they obtained 4.9% accuracy gain for the TIMIT phoneme recognition and more than half error reduction for the TI connected digits recognition. De Haan and Ececioglu reported 97.2% recognition accuracy for the TI isolated digits by using a 19-dimensional feature vector and a feature map strategy [5]. Another interesting result is 50.6% recognition accuracy for TIMIT phoneme recognition reported by Pepper and Clements, who employed 25-dimensional feature vectors and a large 128-state HMM [14].

VI. CONCLUSIONS

We have evidence that most phonemes reveal Gaussian distribution characteristics and that the phoneme-based Gaussian mixture modeling is an efficient method for vector quantization in a DHMM speech recognition system. Improvements have been achieved by using the EM algorithm as an alternative method for VQ, while the phoneme-based VQ leads to noticeable improvements in recognition of a DHMM system. The phoneme-based VQ-HMM system can be used for phoneme recognition or isolated word recognition. The recognition accuracies can be increased by use of a multiple-featured speech vector or multiple VQ code books.

Further investigation of the speech space is needed to more accurately describe the temporal as well as stationary partitioning of the speech space. Another important extension of this work is segmentation of phonemes and subword level segmentation of utterances.

ACKNOWLEDGMENT

The authors would like to thank Dr. C. deSilva for his help in the programming of the EM algorithm and the projection program.

REFERENCES


Yaxin Zhang (S’95) received the M.S. degree from the Institute of North-West Telecommunication Engineering, Xian, China, in 1987, and the Ph.D. degree in electrical engineering from The University of Western Australia, Perth, Australia, in 1996.

After 1987, he was a lecturer at Shaanxi Institute of Technology, Shaanxi, China. In 1990, he joined the Centre for Intelligent Information Processing Systems, The University of Western Australia, as a visiting researcher. From 1992 to 1995, he studied at the Centre as a Ph.D. candidate. Currently, he is a researcher at Motorola Research Centre, Bontany, Australia. His research interests include speech signal processing and automatic speech recognition.

Roberto Togneri (S’87–M’88) received the Ph.D. degree in electrical engineering from The University of Western Australia, Perth, Australia, in 1989.

He is a Lecturer with the Department of Electrical and Electronic Engineering, The University of Western Australia. His research interests include speech signal processing, speech recognition, and artificial neural networks for clustering and classification, as well as computer networking and parallel computations over workstation clusters.

Dr. Togneri is currently Vice-Chair of the IEEE Section Committee, and is on the Executive Committee of the Australian Speech Science and Technology Association (ASSTA).

Michael Alder (A’89) received the B.S. degree in physics from Imperial College, University of London, in 1963, the Ph.D. in pure mathematics from Liverpool University in 1972, and the Master Eng. degree from The University of Western Australia, Perth, Australia, in 1988.

Currently, he is a lecturer with the Department of Mathematics at The University of Western Australia. His research interests include the use of artificial neural networks for pattern recognition.