http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (1): 107-115, 2010
ISSN 1812-5638

Improving Speaker Verification in Noisy Environments using Adaptive Filtering and Hybrid Classification Technique

M.Z. Ilyas, S.A. Samad, A. Hussain and K.A. Ishak
Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: This study describes two approaches of improving speaker verification in noisy environments. The first approach is implementation of a speaker verification classification technique base on hybrid Vector Quantization (VQ) and Hidden Markov Models (HMMs) in clean and noisy environments. The second approach is implementation of Adaptive Noise Cancelation (ANC) as pre-processing for noise removal. The motivation to implement hybrid classification technique is to improve the HMMs performance. It is shown that, by using the hybrid technique, an Equal Error Rate (EER) of 11.72% is achieved compared to HMM alone, which achieved 16.66% in clean environments. However, both techniques show degradation in noisy environments. In order to address these problems, an Adaptive Noise Cancellation (ANC) technique using adaptive filtering is implemented in the pre-processing stage due to its ability to separate overlapping speech frequency bands. Investigations using Least-Mean-Square (LMS), Normalized Least-Mean-Square (NLMS) and Recursive Least-Squares (RLS) adaptive filtering are conducted to find the best solution for the speaker verification system.

Key words: Vector quantization, hidden Markov models, adaptive filtering, least mean-square, normalized least mean-square, recursive least squares

INTRODUCTION

Biometrics is the study of automated methods of recognizing a person based on measurable physiological or behavioral characteristics (NSTC, 2006a). Biometric recognition systems are in demand today because of their reliance on human features that are unique to each person and cannot be forged easily, such as face, fingerprint, hand geometry, handwriting, iris, retina, vein and voice. Speaker recognition or verification is a biometric modality that uses an individual's voice for recognition or verification purpose. It is a different technology from speech recognition, which recognizes words as they are articulated (NSTC, 2006b). Speech contains many characteristics that are specific to each individual. For this reason, listeners are often able to recognize the speaker's identity fairly quickly even without looking at the speaker. Speaker verification is a process of determining whether a person is who he or she claims to be by using his or her voice (Campbell, 1997; Naik, 1990; Rabiner and Juang, 1993; NSTC, 2006b).

Many classification techniques have been proposed for speaker verification systems including Dynamic Time Wrapping (DTW) (Al-Haddad *et al*., 2008), Hidden Markov Models (Han *et al*., 2003; Rabiner, 1989; Yoshizawa *et al*., 2004), Artificial Neural Networks (ANNs) and Vector Quantization (VQ) (Linde *et al*., 1980; Soong *et al*., 1985; Vasuki and Vanathi, 2006) and Support Vector Machine (SVM) (Wan and Renals, 2005). The most popular classification technique in speaker verification is HMMs (NSTC, 2006b; Furui, 1997; Rabiner and Juang, 1993). In particular, recognition or verification systems based on HMMs are effective under many circumstances, but they suffer from major limitations that limit applicability. Generally, the performance of the single technique is limited. Applications that require high security such as internet banking and high security access control can be obtained by using hybrid techniques.

This study presents a hybrid VQ and HMMs classification technique. The goal in hybrid technique for speaker verification system by using VQ and HMMs is to take advantage of the properties of both VQ and HMMs, improve flexibility and verification performance. Other hybrid architectures that can be found in the literature are hybrid ANN/HMM (Trenti and Gori, 2000), hybrid TDNN/HMM (Jang and Un, 1996), hybrid MMI-connectionist/HMM (Neukirchen and and Rigoll,

Corresponding Author: M.Z. Ilyas, Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia Tel: 60-03-9621-6317 Fax: 60-03-8921 6146

1997) and combining SDTW and independent classification (Lichtenaur *et al*., 2008). However, this study focuses on improving HMMs by using a hybrid technique with VQ in speaker verification and the performance between other hybrid techniques are not compared due to the difference in the data set used.

In most real world applications, the speech from speakers is captured in non-ideal situations such as in noisy environments which may seriously reduce system performance (Fujimoto and Ariki, 2000; Hussain *et al*., 2007). Over several decades, a significant amount of research attention has been focused on the signal processing techniques that are able to extract a desired speech signal and reduce the effects of unwanted noise. Depending on the number of sensors used in the system, these approaches can classified into three basic categories, namely temporal filtering techniques using only a single microphone, adaptive noise cancellation utilizing a primary sensor to pick up the noisy signal and reference sensor to measure the noise field and beam forming techniques exploiting an array of sensors (Yiteng *et al*., 2006).

In this study, we propose a novel approach by using hybrid VQ and HMMs classification technique together with adaptive noise cancellation based on LMS, NLMS and RLS adaptive filter. We also compare the performance of the adaptive filter to select the best filter for the systems. The objective is to improve the performance of a speaker verification system for both clean and noisy environments. The technique is evaluated using a Malay spoken digit database for clean environment and Gaussian white noise is added to the data to evaluate the system performance for noisy environments.

SPEAKER VERIFICATION

Hidden Markov Models (HMMs): A speaker verification system consists of two phases: training phase and verification. In the training phase, the speaker voices are recorded and processed in order to generate its model to store in the database. While, in the verification phase, the existing reference templates are compared with the unknown voice input. In this study, the HMM method is used as the training algorithm.

The most flexible and successful approach to speech recognition so far has been HMMs. The goal of HMMs parameter estimation is to maximize the likelihood of the data under the given parameter setting. General theory of HMMs has been given by Rabiner and Juang (1986, 1993). There are 3 basic parameters in HMMs which are:

- π: The initial state distribution
- a: The state-transition probability matrix
- b: Observation probability distribution

In the training phase, a HMM model for each speaker is generated. Each model is an optimized model for the word it represents. For example, a model for the Malay word Satu (number one), has its a, b and π parameters adjusted so as to give the highest probability score whenever the word 'Satu' is uttered and lower scores for other words.

A training set is needed to build a model for each speaker. This training set consists of sequences of discrete symbols, such as the codebook indices obtained from the VQ stage. Here, an example is given of how HMMs are used to build models for a given training set. Assuming that N speakers are to be verified, first we must have a training set of L token words and an independent testing set. These are the steps needed during speaker verification process:

- First we build a HMM model for each speaker. The L training set of tokens for each speaker will be used to find the optimum parameters for each word model. This is done using the re-estimation formula
- Then, for each unknown speaker in the testing set, first characterize the speech utterance into an observation sequence. This means using an analysis method for the speech utterance so that we get the feature vector and then the vector is quantized using VQ. Thus, we will get a sequence of symbols, with each symbol representing the speech feature for every discrete time step
- We calculate a, b and π parameters for the observation sequence using one of the speaker models in the vocabulary. Then repeat for every speaker model in the database

After N models have been created, the HMM engine is then ready for speaker verification. A test observation sequences from an unknown speech utterance produced after vector quantization of cepstral coefficient vectors, is evaluated using the Viterbi algorithm. The log-Viterbi algorithm is used to avoid precision underflow. For each speaker model, probability score for the unknown observation sequence is computed. The speaker whose model produces the highest probability score and matches the ID claimed is then selected as the client speaker.

Speaker verification means making a decision on whether to accept or reject a speaker. To decide, a threshold is used with each client speaker. If the unknown speaker's maximum probability score exceeds this threshold, then the unknown speaker is verified to be the

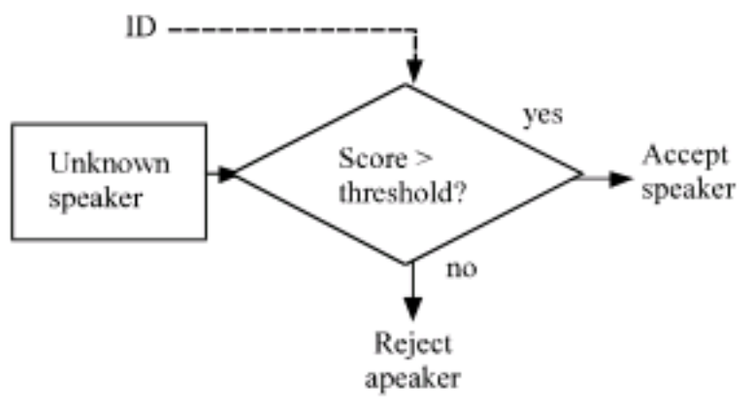

Fig. 1: Speaker verification decision

client speaker. However, if the unknown speaker's maximum probability score is lower than this threshold, then the unknown speaker is rejected. The relationship is shown in Fig. 1.

The threshold is determined as follows:

- For each speaker, evaluate all samples spoken by him using his own HMM models and find the probability scores. From the scores, find the mean, μ_1 and standard deviation, σ_1, of the distribution
- For each speaker, evaluate all samples spoken by a large number of impostors using the speaker's HMMs models and find the probability scores. From the scores, find the mean μ_2 and standard deviation σ_2 of the distribution
- For each speaker, calculate the threshold as:

$$T = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} \quad (1)$$

Vector quantization (VQ): VQ is a process of mapping vectors of a large vector space to a finite number of regions in that space. Each region is called a cluster and is represented by its centre (called a centroid) (Soong *et al*., 1985; Vasuki and Vanathi, 2006). A collection of all the centroids makes up a codebook. The amount of data is significantly less, since the number of centroids is at least ten times smaller than the number of vectors in the original sample. This will reduce the amount of computations needed for comparison in next stages. Even though the codebook is smaller than the original sample, it still accurately represents a person's voice characteristics. The only difference is that there will be some spectral distortion.

In the feature extraction stage, we calculate the LPC cepstrum and the entire speech signal are represented as the LPC to cepstrum parameters and a large sample of these parameters are generated as the training vectors. During the training process of VQ, a codebook is obtained from these sets of training vectors. These training vectors are actually compressed to reduce the storage requirement. An element in a finite set of spectra in a codebook is called a codevector. The codebooks are used to generate indices or discrete symbols that will be used by the discrete HMMs. Hence, data compression of speech is accomplished by VQ in the training phase and the encoding phase that finds the input vectors the best codevectors.

To implement VQ, first, we must get the codebook. A large set of spectral analysis vectors (or speech feature vectors) is required to form the training step. If we denote the size of the VQ codebook as $M = 2^N$ codewords, then we require an L (with L >> M) number of training vectors. It has been found that L should at least be 10 M in order to train a VQ codebook that works well. For this research, we use the LBG algorithm, also known as the binary split algorithm.

The speaker verification based VQ codebook generation proposed by Soong *et al*. (1985) can be summarized as follows:

Given a set of I training features vectors, $(a_1, a_2, ..., a_I)$ characterizing the variability of speaker, we want to find a partitioning of the feature vector space, $(S_1, S_2, \ldots, S_M)$, for that particular speaker where, S, the whole feature space is represented as $S = S_1 \cup S_2 \cup \ldots \cup S_M$. Each partition, S_i, forms a convex, non-overlapping region and every vector inside S_i is represented by the corresponding centroid vector, b_i, of S_i. The partitioning is done in such way the average distortion is minimized over the whole training set:

$$D = \frac{1}{I}\sum_{i=1}^{I} \min_{1 \le j \le M} d(a_i, b_j) \quad (2)$$

The distortion between the vectors a_i and b_j is denoted as $d(a_i, b_j)$. Short-time LPC vectors are used as feature vectors. The corresponding distortion measure to measure the similarity between any two features vectors is the LPC likelihood ratio distortion measure. The likelihood ratio distortion between two LPC vectors a and b is defined as:

$$d_{LR}(a,b) = \frac{b^T R_a b}{a^T R_a a} - 1 \quad (3)$$

where, R_a is the autocorrelation matrix of speech input data associated with the vector a. Using this distortion measure and the VQ codebook training algorithm proposed by Linde, Buzo and Gray (LBG) (Linde *et al*., 1980). We generated speaker-based VQ codebooks. The input speech signal is sampled, segmented and LPC analyzed giving sequence of vectors $a_1, a_2, ..., a_I$. The resultant LPC vector are vector quantized using the N

codebooks corresponding to the N different speakers. The quantization errors (distortion) with respect to each codebook are individually accumulated across the whole test token. The average distortion with respect to the ith codebook (speaker) is:

$$D^i = \frac{1}{L}\sum_{l=1}^{L} \min_{1\le j\le M} d(a_l, b_j^i) \tag{4}$$

The N resultant average distortions are compared to find the minimum. The final speaker recognition decision is given by:

$$i^* = \arg\min_{1\le i\le N} D^i \tag{5}$$

A speaker verification system has a similar structure except only the codebook of the claimed identity is used and the resultant average distortion is compared with a present threshold to reject or to accept the identity claim made by unknown speaker.

ADAPTIVE NOISE CANCELLATION (ANC)

Conventional frequency-selective digital filters with fixed coefficients are designed to have a given frequency response chosen to alter the spectrum of the input signal in a desired manner. However, there are many practical application problems that cannot be successfully solved by using fixed digital filters because either we do not have sufficient information to design a digital filter with fixed coefficients or the design criteria change during the normal operation of the filter (Manolakis *et al.*, 2000).

The principle of general noise cancellation is illustrated in Fig. 2. The s(n) signal is corrupted by uncorrelated additive noise $v_1(n)$ and the combined signal $s(n)+v_1(n)$ provides a primary input. A second sensor located at a different point, acquires a noise $v_2(n)$ that is uncorrelated with the signal s(n) but correlated with the noise $v_1(n)$. If we can design a filter that provides a good estimate y(n) of the noise $v_1(n)$, by exploiting the correlation between $v_1(n)$ and $v_2(n)$, then we could recover the desired signal by subtracting $y(n)\approx v_1(n)$ from the primary input. The filtered signal is given by estimation error:

$$e(n) = s(n) + [v_1(n) - y(n)] \tag{6}$$

where, y(n) depends on the filter structure and parameters. The mean square error (MSE) is given by:

$$E\{|e(n)|^2 = E\{|s(n)|^2\} + E\{|v_1(n) - y(n)|^2\} \tag{7}$$

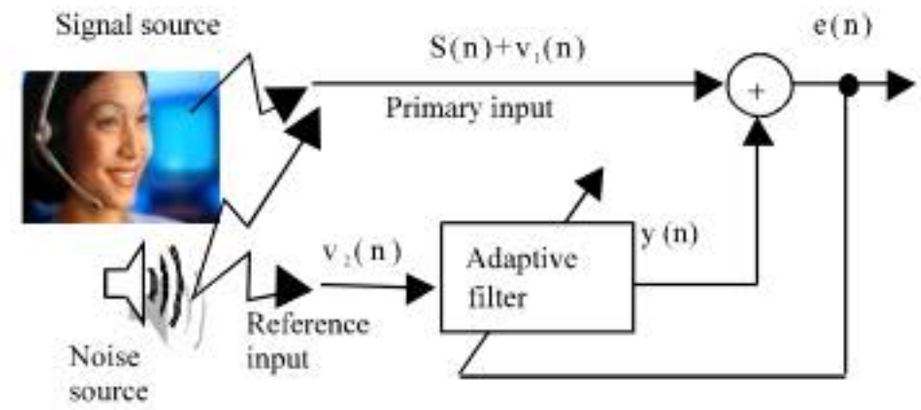

Fig. 2: Adaptive noise cancellation using reference input

However, the performance of the ANC is highly dependent on the quality of the noise reference. The noise in the reference sensor and the noisy speech sensor must be sufficiently correlated to obtain substantial noise reduction. Any leakage from the primary speech signal into the noise reference signal must be avoided since it results in the primary speech signal distortion and poor noise cancellation.

Least mean square (LMS) adaptive filter: The LMS algorithm is an important member of the family of the stochastic gradient algorithms. A significant feature of the LMS algorithm is its simplicity. Moreover, it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Indeed, it is the simplicity of the LMS algorithm that has made it the standard against which other linear adaptive filtering algorithms are benchmarked (Haykin, 2002). Block diagram of adaptive transversal filter is illustrated in Fig. 3.

The output vector of LMS adaptive filter is given by:

$$\hat{y}(n) = \hat{w}^H(n)\hat{u}(n) \tag{8}$$

where, $\hat{w}(n)$ is current estimate of tap-weight vector and $\hat{u}(n)$ is tap-input vector. Further, the superscript H stands for Hermitian, or equivalently conjugate transpose. Estimation error signal is found as:

$$e(n) = d(n) - \hat{y}(n) \tag{9}$$

where, d(n) is desired response. The LMS algorithm is defined as:

$$\hat{w}(n+1) = \hat{w}(n) + \mu\hat{u}(n)e^*(n) \tag{10}$$

where, $\hat{w}$ (n+1) is the estimate of tap-weight vector at time (n+1) and (n+1) is a constant:

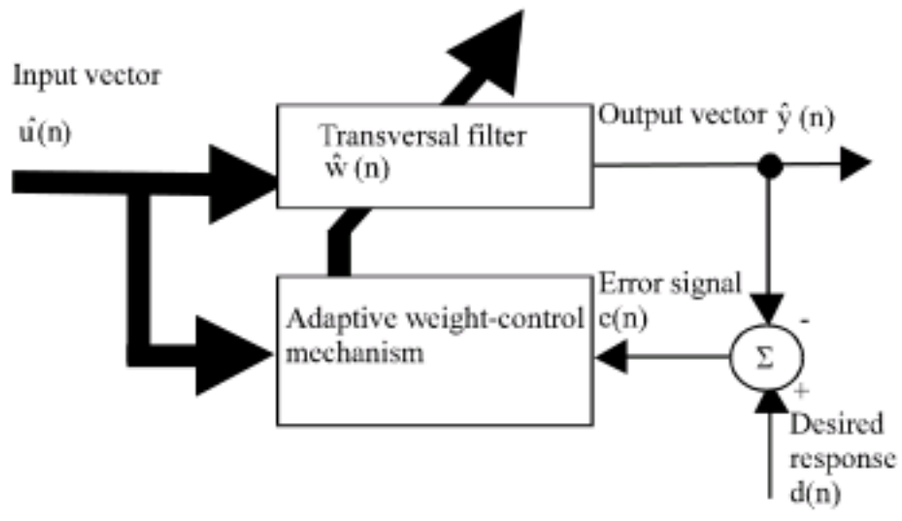

Fig. 3: Block diagram of adaptive transversal filter

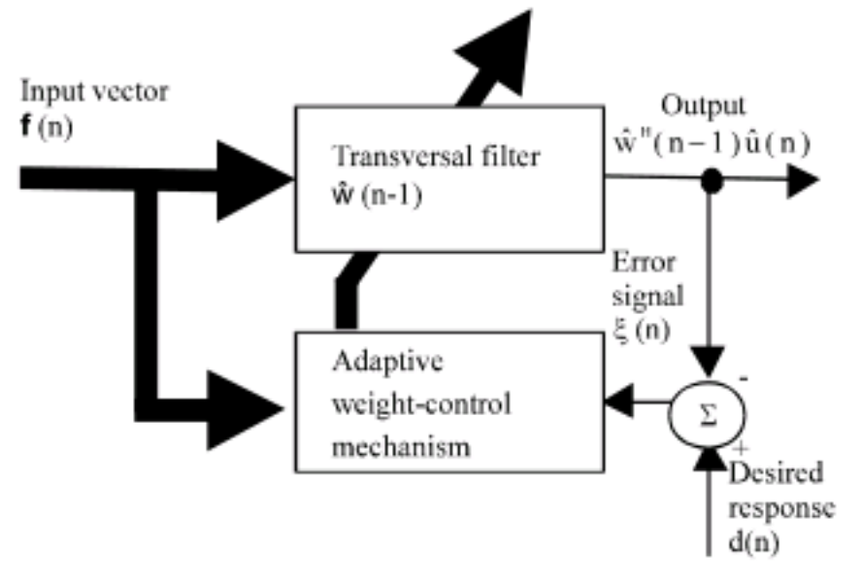

Fig. 4: Block diagram of RLS filter

Normalized least mean square (NLMS) adaptive filter: NLMS filter is in the family of LMS filter. The NLMS filter differs from the conventional LMS in the way in which the step size for controlling the adjustment to the filter's tap-weight vector is defined. The principle of the NLMS adaptive filters in structural terms is exactly the same as the standard LMS filter, as shown in the Fig. 3. Both adaptive filters are built around a transversal filter, but differ only in the way in which the weight controller is mechanized. The M-by-1 tap-input vector $\hat{u}(n)$ produces an output $\hat{y}(n)$ that is subtracted from the desired response d(n) to produce the estimation error e(n). In response to the combined action of the input vector $\hat{u}(n)$ and error signal e(n), the weight controller applies a weight adjustment to the transversal filter. This sequence of events is repeated for a number of iterations until the filter reaches steady state. The output vector of NLMS adaptive filter is given by:

$$\hat{y}(n) = \hat{w}^H(n)\hat{u}(n) \tag{11}$$

where, $\hat{w}(n)$ is current estimate of tap-weight vector and $\hat{u}(n)$ is tap-input vector. Further, the superscript H stands for Hermitian, or equivalently conjugate transpose. Estimation error signal is found as:

$$e(n) = d(n) - \hat{y}(n) \tag{12}$$

where, d(n) is desired response. The NLMS algorithm is defined as:

$$\hat{w}(n+1) = \hat{w}(n) + \frac{\bar{\mu}}{\delta + \|\hat{u}(n)\|^2}\hat{u}(n)e^*(n) \tag{13}$$

where, $\hat{w}(n+1)$ is the estimate of tap-weight vector at time (n+1), $\bar{\mu}$ and δ are constants ($\delta>0$) (Haykin, 2002; Manolakis *et al.*, 2000).

Recursive Least-Squares (RLS) adaptive filter: The RLS algorithm is based on the least-squares (LS) estimate of the filter coefficients $\hat{w}(n-1)$ at iteration (n-1), by computing its estimate at iteration n using the newly arrived data. The filter coefficients at time n are chosen to minimize the cost function:

$$E(n) = \sum_{i=1}^{n} \lambda^{n-i}|e(i)|^2 \tag{14}$$

where, the error signal e(i) is computed for all times $1 \le i \le n$ using the current filter coefficients $\hat{w}(n)$: $e(i) = d(i) - \hat{w}^H(n)\hat{u}(i)$ and λ is called the forgetting factor. Using a matrix inversion lemma a recursive update equation for $\hat{p}(n) = \Phi^{-1}(n)$ is found as:

$$\hat{p}(n) = \lambda^{-1}\hat{p}(n-1) - \lambda^{-1}\hat{k}(n)\hat{u}^H(n)\hat{p}(n-1) \tag{15}$$

with

$$\hat{k}(n) = \frac{\lambda^{-1}\hat{p}(n-1)\hat{u}(n)}{1 + \lambda^{-1}\hat{u}^H\hat{p}(n-1)\hat{u}(n)} \tag{16}$$

Finally, the weight update equation is:

$$\hat{w}(n) = \hat{w}(n-1) + \hat{k}(n)\xi^*(n) \tag{17}$$

where, $\xi(n)$ is the a priori estimation error and is given by:

$$\xi(n) = d(n) - \hat{w}^H(n-1)\hat{u}(n) \tag{18}$$

Equation 18 describes the filtering operation of the algorithm, whereby the transversal filter is excited to compute the a priori estimation error $\xi(n)$. Equation 17 describes the adaptive operation of the algorithm, whereby the tap-weight vector is updated by incrementing its old value by an amount equal to the product of the complex conjugate of the priori estimation error $\xi(n)$ and time-varying gain vector $\hat{k}(n)$. Equation 15 and 16 enable the value of the gain vector itself. Figure 4

shows the block diagram of the RLS adaptive filter (Haykin, 2002; Manolakis *et al*., 2000; Poularikas and Ramadan, 2006).

EXPERIMENTS

Experimental conditions in clean environment: Speaker verification experiments were carried out using a Malay spoken digit database (Ilyas *et al*., 2007) which contains 100 speakers. To evaluate performance of the system in noisy environments, experiments using added Gaussian white noise at 4 levels (30, 20, 10 and 0 dB) were carried out with and without adaptive filtering. For the experiments, 100 speakers were selected where each speaker has 10 repetitions of Malay digits. All of the Malay digits, from 0 until 9 were selected to build the speaker model. Feature vectors composed of 14 linear predictive coding cepstral (LPCC) coefficients were used. The analyzed frame was windowed by a 15 msec Hamming window with 5 msec overlapping. The samples were pre-segmented automatically using the start-end detection module to remove the silent parts. For speaker modeling, all samples were selected from each speaker's training set. This procedure was for building the global codebook to be used later for HMMs. Then, for each speaker, a codebook was built using the Linde-Buzo-Gray (LBG) VQ method. The size of each codebook was 256 codevectors as for the global codebook. For testing we used a workstation, equipped with a Pentium D processor, with 1 GB of memory and running on the Windows XP operating system.

Figure 5 shows the flow chart of the speaker verification experiment. First, clean speech signals passes through end point detection and feature extraction without adaptive filtering. If it is a training process, it will generate individual codebook for VQ models and global codebook for HMMs models. Otherwise, evaluation of combination VQ and HMMs and standalone HMMs will be conducted based on individuals models and ID.

Experimental conditions in noisy environment: Experiments in noisy environments were carried out using the same approach as in a clean environment. However, Gaussian white noise was added to clean speech signals to produce noisy speech signals. Figure 6 shows an example of original clean signal and noisy speech signals mixed with Gaussian white noise in SNR of 0 dB.

Experimental conditions in noisy environments using adaptive filter: Adaptive filters could be used to cancel or clean the created noisy signal. Filtered signals were tested using the same procedure as clean and noisy environments to evaluate speaker verification system performance. However, noisy speech signals went through adaptive filtering (Fig. 5) before end point detection and feature extraction. Figure 7a-c shows an example of the filtered signal that is obtained from LMS, NLMS and RLS adaptive filtering at SNR of 0 dB (Fig. 6a, b) of speech signal. All of the filtered signals are almost similar with the original clean signal.

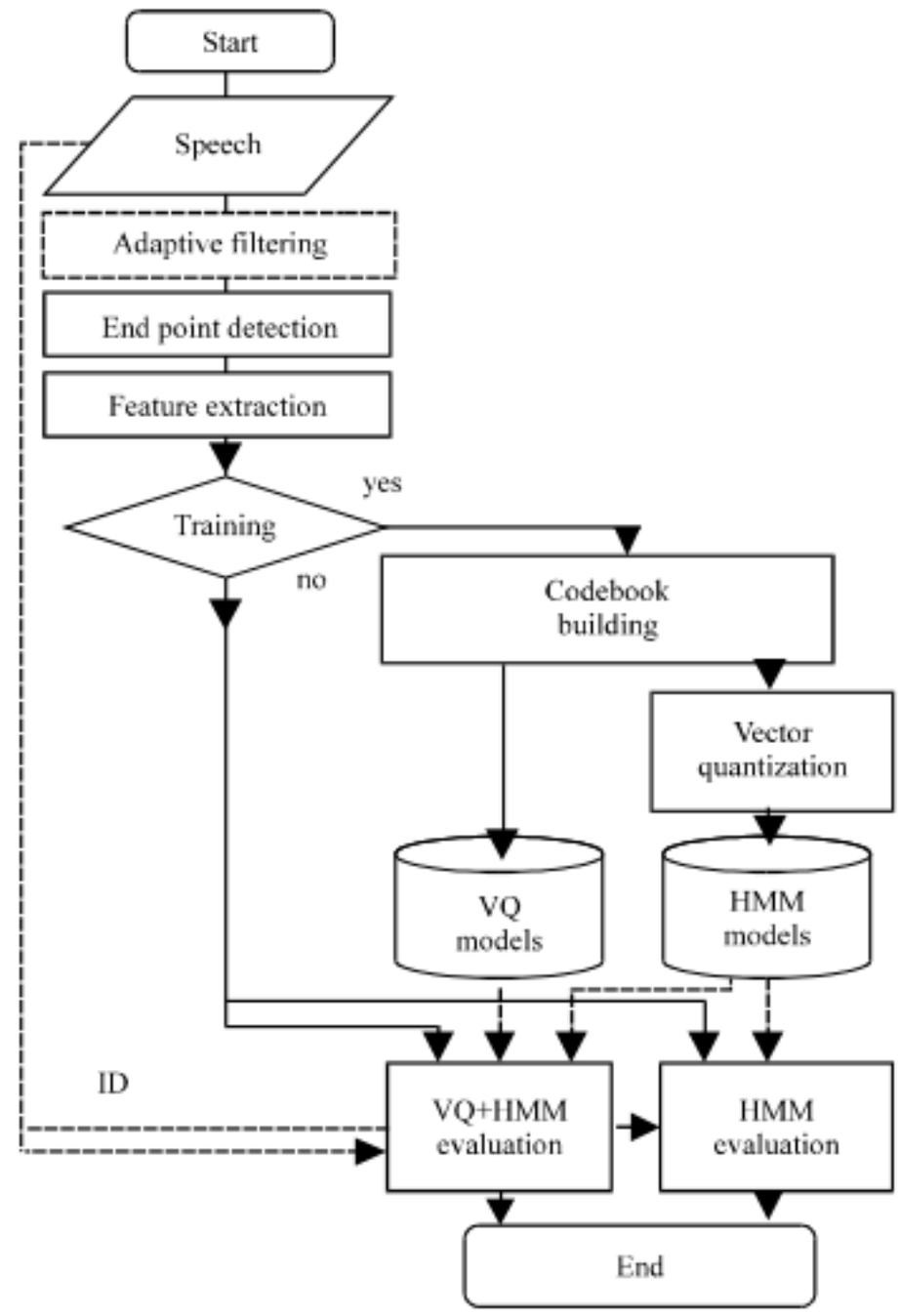

Fig. 5: Speaker verification experiment flow chart

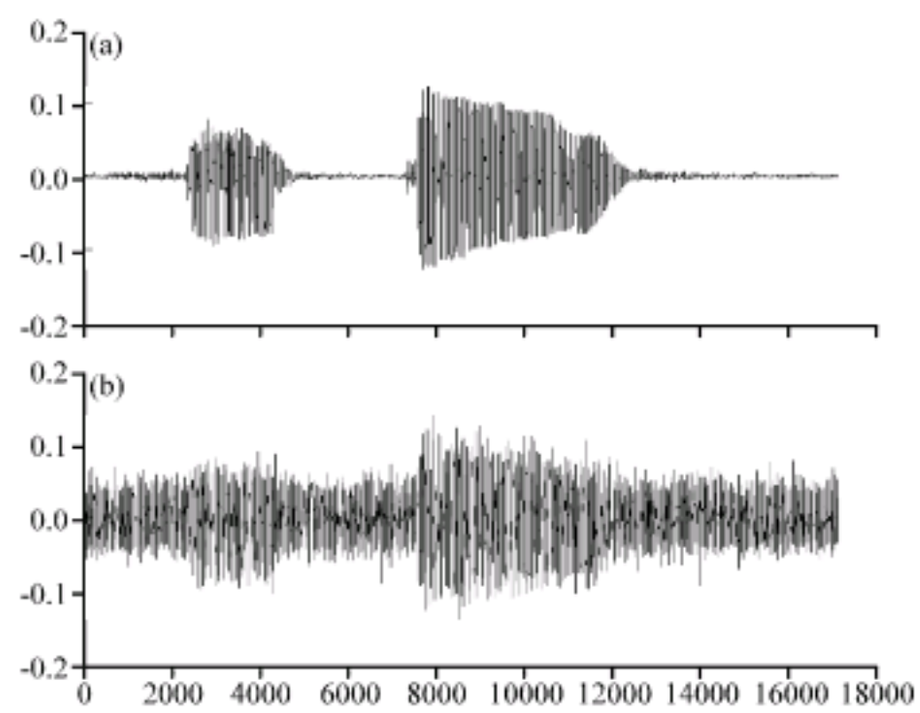

Fig. 6: (a, b) Original and noisy speech signal of the word 'Satu'

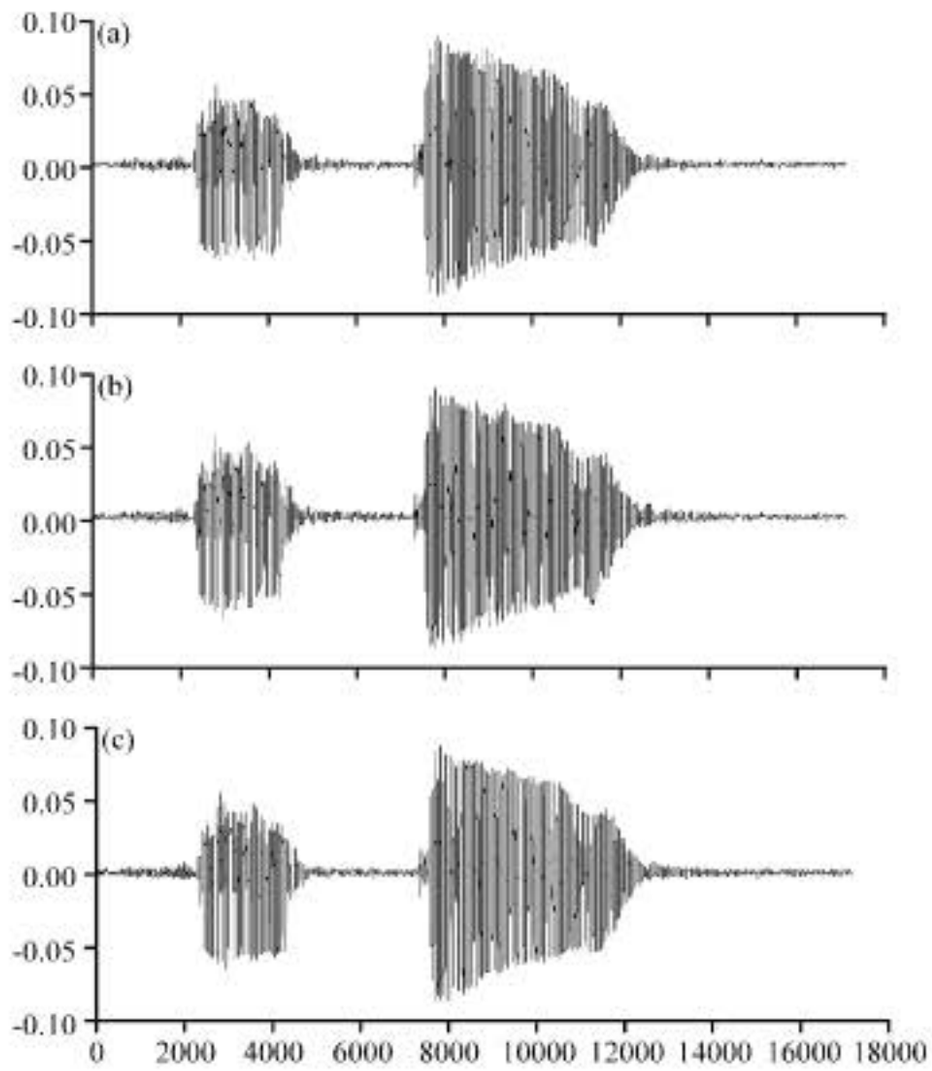

Fig. 7: Filtered speech signal of word 'Satu' with (a) LMS, (b) NLMS and (c) RLS filtered (0 dB)

RESULTS AND DISCUSSION

Clean environment: Table 1 shows a summary of the verification results for the experiments performed. An Equal Error Rate (EER) of 11.72% is achieved using this combination technique compared to stand alone HMM which is 16.66%. Using the combination technique, true speaker rejection rate is 0.06% while impostor acceptance rate is 0.03%. Figure 8 shows a ROC plot of False Rejection Rate (FRR) vs False Acceptance Rate (FAR). It shows that the hybrid technique of VQ and HMMs outperform the HMM only based technique.

Noisy environments (without adaptive filtering): Table 2 shows the verification results of HMMs in noisy environments (Gaussian white noise mixed) and without adaptive filtering. EERs of between 41.01-49.94% are achived for SNRs of between 0-30 dB. High noise levels worsen the system performance in all cases. Table 3 shows the verification results of hybrid VQ+HMMs in noisy environments (Gaussian white noise mixed) without adaptive filtering. EERs of between 37.14-49.11% are achived for SNRs of between 0-30 dB. Using the hybrid technique, a relative improvement of EER between 0.83-3.87%, FAR of between 19.92-26.43% and FRR of between 4.44-24.55% are obtained compared to HMMs technique.

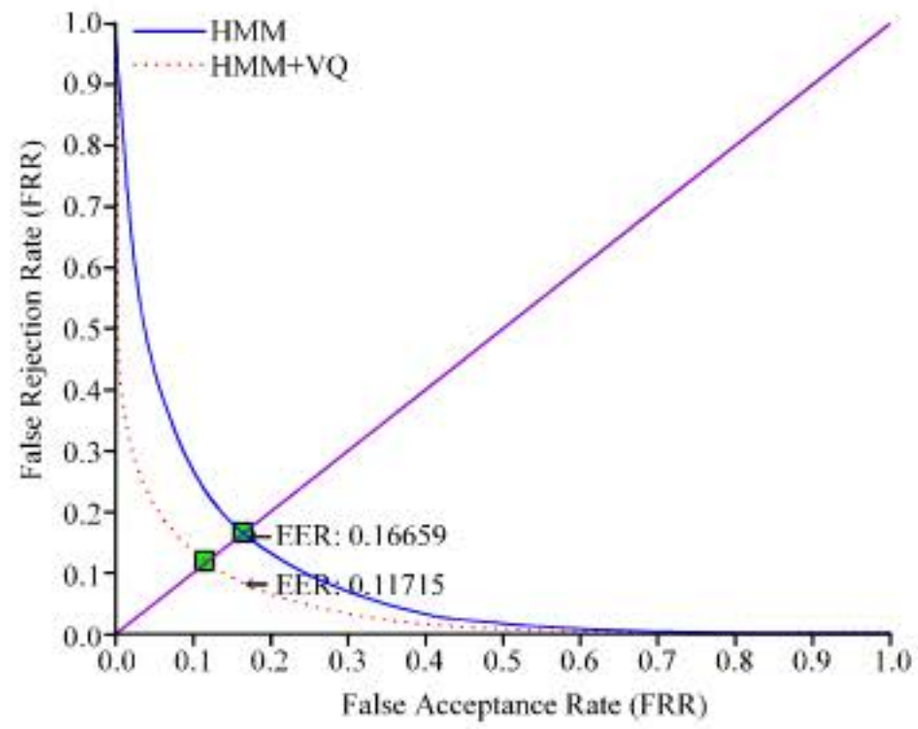

Fig. 8: ROC plot of False Rejection Rate (FRR) vs. False Acceptance Rate (FAR)

Table 1: Verification result for clean environment (%)

SNR (dB)	Method	FRR	FAR	EER
Clean	HMM	25.30	9.99	16.66
	VQ+HMM	0.06	0.03	11.72

Table 2: Verification result of HMMs in noisy environments (%)

SNR (dB)	Method	FRR	FAR	EER
0	HMMs (without adaptive filtering)	34.47	57.15	49.94
10		31.52	57.10	48.19
20		28.12	55.42	45.21
30		25.88	48.95	41.01

Table 3: Verification result of VQ+HMMs in noisy environments (%)

SNR (dB)	Method	FRR	FAR	EER
0	VQ+HMMs (without adaptive filtering)	59.02	37.23	49.11
10		51.33	35.38	46.36
20		40.78	28.99	42.35
30		30.32	23.14	37.14

Table 4: Verification result using LMS adaptive filter (%)

SNR (dB)	Method	FRR	FAR	EER
0	VQ+HMMs (with LMS)	29.78	22.62	36.14
10		28.94	24.64	36.30
20		30.19	23.06	37.03
30		37.77	27.44	41.02

Table 5: Verification result using NLMS adaptive filter (%)

SNR (dB)	Method	FRR	FAR	EER
0	VQ+HMMs (with NLMS)	26.44	21.83	34.56
10		26.10	21.22	34.24
20		26.19	21.09	34.17
30		26.23	21.07	34.17

Noisy environments (with adaptive Filtering): Table 4-6 show the verification results using the hybrid VQ and HMMs in noisy environments (Gaussian white noise mixed) and with LMS, NLMS and RLS adaptive filtering, respectively. Figure 9 shows the ROC plot of FAR vs FRR using hybrid VQ and HMMs and with and without adaptive filtering at noisest 0 dB condition. Improvements of 26.41, 29.63 and 31.5% are achieved using LMS, NLMS

Table 6: Verification result using RLS adaptive filter (%)

SNR (dB)	Method	FRR	FAR	EER
0	VQ+HMM (with RLS)	25.17	21.26	33.64
10		27.27	22.93	35.37
20		29.76	22.69	36.49
30		31.60	24.57	37.83

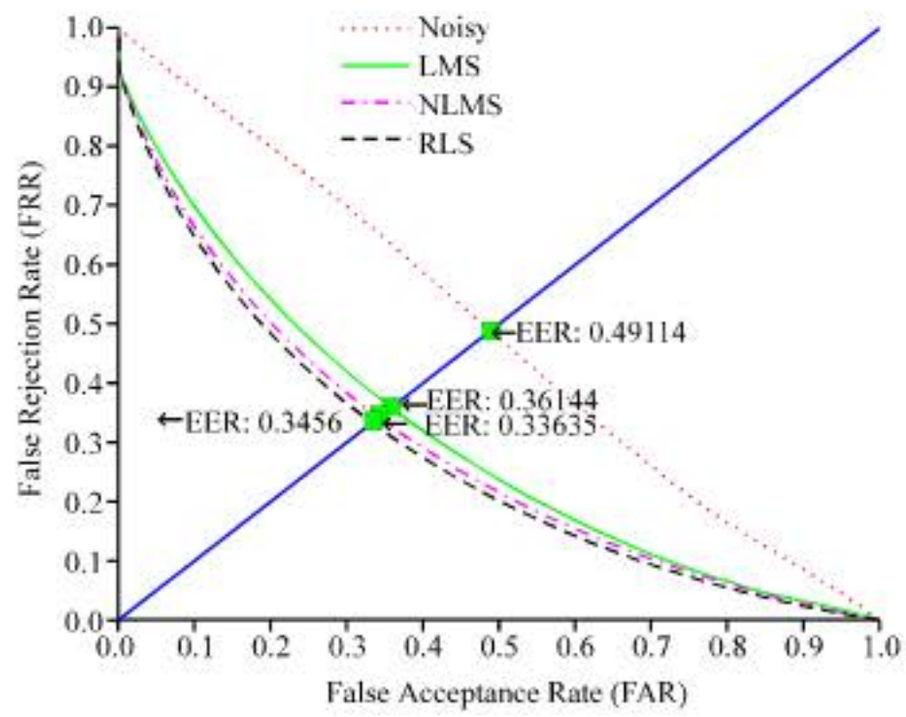

Fig. 9: ROC plot of False Rejection Rate (FRR) vs. False Acceptance Rate (FAR) with different of adaptive filter at 0 dB SNR

and RLS respectively. It can be seen that RLS adaptive filter gives the best result for the nosiest (0 dB) condition in terms of speed of adaptation and speech tracking behavior. However, as far as computational complexity is concerned, RLS algorithm implies that complexity increases with the squared of filter order (ON^2), where N is the filter order. On the other hand, the LMS algorithm gives the lowest computational requirements since the complexity of such an algorithm is directly proportional to the filter order N. The NLMS algorithm has variable step size for adaptation which posses a better tracking characteristics with same computational complexity as the LMS version.

The traditional HMMs perform worse in both clean and noisy environment compared to hybrid VQ and HMMs. Although hybrid VQ and HMMs perform better in both clean and noisy environment, its performance still degrades under noisy condition. Implementation of the adaptive filtering improves the hybrid VQ and HMMs in noisy condition. The method proposed in the paper is reasonable with the following justification:

- Recognition systems based on HMMs are effective under many circumstances, but do suffer from some major limitations that limit applicability of automatic speech recognition (ASR) technology in real-world environments (Trenti and Gori, 2000)
- The goal in hybrid systems in ASR (speaker verification) is to take advantage from the properties of both HMMs and ANNs (VQ) (Trenti and Gori, 2000)
- The adaptive filter relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in a environment where complete knowledge of the relevant signal characteristics is not available (Haykin, 2002)

CONCLUSION

This study has shown two approaches of improving speaker verification in clean and noisy environments. The first approach show how hybrid VQ and HMMs improves a HMMs speaker verification performance in clean and noisy environments and the second approach shows how adaptive filtering improves the hybrid technique in noisy environments. Experimental results have shown that by using this hybrid classification technique, EER, FAR and FRR are improved in clean environment and noisy environments compared to HMMs alone. However, both techniques shown degradation in noisy environments. In order to address these problems, an Adaptive Noise Cancellation (ANC) technique using adaptive filtering is implemented due to its ability to separate overlapping speech frequency bands. Investigations using Least-Mean-Square (LMS), Normalized Least-Mean-Square (NLMS) and Recursive Least-Squares (RLS) adaptive filtering are conducted to find the best solution for the system. It has been shown that RLS adaptive filter gives the best result for the nosiest (0 dB) condition. However, considering computational complexity and overall results, NLMS adaptive filter is identified as the best filter. Further work will require concentration on real-time noisy conditions.

ACKNOWLEDGMENTS

This research is supported by the following research grant: Fundamental Research Grant Scheme, Malaysian Ministry of Higher Education, FRGS UKM-KK-02-FRGS-0036-2006 and E-science, Ministry of Science Technology and Innovation, e-science 01-01-02-SF0374.

REFERENCES

Al-Haddad, S.A.R., S.A. Samad, A. Hussain and K.A. Ishak, 2008. Isolated Malay digit recognition using pattern recognition fusion of dynamic time warping and hidden Markov models. Am. J. Applied Sci., 5: 714-720.

Campbell, J.P., 1997. Speaker recognition: A tutorial. Proc. IEEE, 85: 1437-1462.

Fujimoto, M. and Y. Ariki, 2000. Noisy speech recognition using noise reduction method based on Kalman filter. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Jun. 5-9, Istanbul, Turkey, pp: 1727-1730.

Furui, S., 1997. Recent advances in speaker recognition. Pattern Recog. Lett., 18: 859-872.

Han, W., K.W. Hon, C.F. Chan, T. Lee, C.S. Choy, K.P. Pun and P.C. Ching, 2003. An HMM-based speech recognition IC. Proceedings of the International Symposium on Circuits and Systems, May 25-28, IEEE Computer Society, Washington, DC, USA., pp: II-744-II-747.

Haykin, S., 2002. Adaptive Filter Theory. 4th Edn., Prentice-Hall, New Jersey, ISBN: 0130901261.

Hussain, A., M. Chetouani, S. Squartini, A. Bastari and F. Piazza, 2007. Nonlinear Speech Enhancement: An Overview. Springer-Verlag Berlin Heidelberg. New York.

Ilyas, M.Z., S.A. Samad, A. Hussain and K.A. Ishak, 2007. Speaker verification using vector quantization and hidden markov model. Proceedings of the 5th Student Conference on Research and Development, Dec. 12-11, Selangor, Malaysia, 1-5.

Jang, C.S. and C.K. Un, 1996. A new parameter smoothing method in the hybrid TDNN/HMM architecture for speech recognition. Speech Commun., 19: 317-324.

Lichtenaur, J.F., E.A. Hendriks and M.J.T. Reinders, 2008. Sign language recognition by combining statistical DTW and independent classification. IEEE. trans. pattern anal. machine intell., 30: 2040-2046.

Linde, Y., A. Buzo and R.M. Gray, 1980. An algorithm for vector quantizer design. IEEE Trans. Commun., 28: 84-95.

Manolakis, D.G., V.K. Ingle and S.M. Kogon, 2000. Statistical and Adaptive Signal Processin: Spectra Estimation, Signal Modeling, Adaptive Filtering and Array Processing. McGraw-Hill, USA.

Naik, J.M., 1990. Speaker verification: A tutorial. IEEE. Commun. Magazine, 28: 42-48.

National Science and Technology Council (NSTC), 2006a. Privacy and biometrics building a conceptual foundation. pp: 1-10.

National Science and Technology Council (NSTC), 2006b. Speaker recognition. pp: 15-17.

Neukirchen, C. and G. Rigoll, 1997. Advanced training methods and new network topologies for hybrid MMI-connectionist/HMM speech recognition systems. Proc. IEEE-ICASSP-97, 4: 3257-3260.

Poularikas, A.D. and Z.M. Ramadan, 2006. Adaptive Filtering Primer with Matlab. CRC. press, Boca Raton.

Rabiner, L.R. and B.H. Juang, 1986. An introduction to hidden Markov models. IEEE. ASSP. Magazine, 3: 4-16.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77: 257-286.

Rabiner, L.R. and B.H. Juang, 1993. Fundamental of Speech Recognition. Prentice Hall, New Jersey.

Soong, F., A. Rosenberg, L. Rabiner and B. Juang, 1985. A vector quantization approach to speaker recognition. Proceedins of the IEEE International Conference on Acoustics, Speech and Signal Processing, Apr. 1985, IEEE Computer Society, Washington, DC, USA., pp: 387-390.

Trenti, E. and M. Gori, 2000. A survey of hybrid ANN/HMM models for automatic speech recognition. J. Neurocomput., 37: 91-125.

Vasuki, A. and P.T. Vanathi, 2006. A review of vector quantization techniques. Potentials IEEE., 25: 39-47.

Wan, V. and S. Renals, 2005. Speaker verification using sequence discriminant support vector machine. IEEE. Trans. audio process., 13: 203-210.

Yiteng, H., B. Jacob and C. Jingdong, 2006. Acoustic MIMO Signal Processing: Signal and Communication Technology. Springer, New York.

Yoshizawa, S., N. Wada, N. Hayasaka and Y. Miyanaga, 2004. Scalable architecture for word HMM-based speech recognition. Proceedings of the International Symposium on Circuits and Systems, May 23-26, IEEE Computer Society, Washington, DC, USA., 417-420.