Embedded speaker verification in low cost microcontroller

Maximiliano Lizondo, Pablo D. Agüero, Alejandro J. Uriz, Juan C. Tulli and Esteban L. Gonzalez
Facultad de Ingeniería
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina
Email: mlizondo@fi.mdp.edu.ar

***Abstract*—Automatic speaker recognition is the use of a machine to identify or verify the identity an individual from a spoken sentence. This paper describes an implementation of a embedded speaker verification system that may be used for an electronic door lock, among other possible applications. The system is built in a low cost dsPIC from Microchip, which combines features of microcontrollers with characteristics of DSPs in a single 16-bit high-performance core. Some special programming techniques used for the implementation are described. The aim was optimizing the code for speed and memory usage. Experimental results in MATLAB of the embedded speaker verification algorithms show promising results, with a false acceptance rate of 8% for a false rejection rate of 12%.**

***Index Terms*—Speaker verification, embedded system, dsPIC.**

I. Introduction

Biometric recognition refers to the use of distinctive characteristics to identify individuals [1], [2]. These biometric identifiers are usually classified into physiological or behavioural characteristics.

Physiological biometrics, like fingerprints, face, hand geometry, retina or iris, are physical characteristics that can be measured at some particular point in time. On the other hand, behavioural biometrics like signature, voice or gait, consist of actions that extend over time. Unlike physiological biometrics, behavioural biometrics are learned or acquired over time and they can be easily and deliberately changed [2].

Speech is one of the most natural modalities of human interaction, a fact corroborated by many years of research and development in speech processing. Recent developments in speech technologies have finally provided truly functional applications. An example of these applications is the role of speech as a biometric identifier for automatic speaker recognition.

Automatic speaker recognition is the use of a machine to identify an individual from an utterance. Recently, this technology has undergone an increasing importance in applications such as access control, transaction authentication, law enforcement, forensics, and system customisation, among others.

One of the central questions addressed by this field is what conveys speaker identity in the speech signal. Traditionally, automatic speaker recognition systems have relied mostly on short-term features related to the spectrum of the voice. However, human speaker recognition relies on other additional sources of information. Therefore, these sources may also play an important role in the automatic speaker recognition task, by adding complementary knowledge to the traditional spectrum-based recognition systems and thus improving their accuracy.

Voice is not expected to have enough distinctiveness to allow the recognition of an individual from a large database of speakers. Moreover, it is characterised by three important disadvantages: first, a speech signal can be degraded in quality by the microphone or the transmission channel; second, voice can be affected by the health of a person, stress or emotions; and finally, it has been shown that some people are extraordinarily skilled in mimicking voices [1], [2]. However, voice is a non-intrusive biometric with a high acceptability. Moreover, it is nowadays the only feasible biometric identifier in applications requiring person recognition over a telephone system [1], [3].

This paper describes an implementation of a embedded speaker verification system that may be used for an electronic door lock, among other possible applications. Our system is built in a low cost dsPIC from Microchip, which combines features of microcontrollers with characteristics of DSPs in a single 16-bit high-performance core.

The paper is organized as follows. Section II briefly describes speaker verification systems, their architecture, feature extraction and statistical models. Section III depicts the proposed embedded speaker verification system, with details of the implementation. Finally, Section IV shows the conclusions and some future directions.

II. Speaker verification system

Depending on the application, an automatic biometric recognition system can run in two modes: identification and verification [4].

In identification mode, the aim is to determine which speaker, in a set of known users (whose models are stored in the database), matches the unknown user. In the verification mode, the aim of a system is to determine whether an unknown user is who he/she claims to be or an impostor. Applications of the later mode are mainly related to access restriction in secured areas, and it is of interest in this paper.

In verification systems (see Fig. 1), a user is claiming an identity. A model corresponding to that identity must be stored in the database, which must contain an impostor model as well. The biometric features of the claimed user are compared to

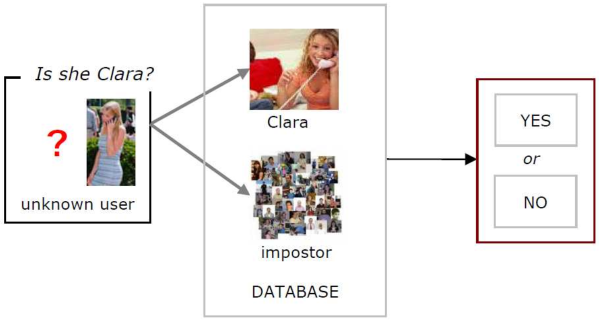

Fig. 1. Automatic speaker verification system.

the model of the claimed identity and to the impostor model. If a user seems to be closer to the claimed identity, he/she will be accepted as a known user. Otherwise, the user will be rejected and treated as an impostor.

After having computed a score of similarity between the input user and the corresponding templates stored in the database, a decision is taken whether the user must be accepted or rejected by the system. However, such decision can be correct or not. If the decision is incorrect, two different errors can occur [3]:

- False rejection: the system rejects a valid identity claim.
- False acceptance: the system accepts an identity claim from an impostor.

Both types of errors give rise to two types of error rates, which are commonly used to measure the performance of a system:

- False rejection rate (FRR): percentage of incorrectly rejected clients.
- False acceptance rate (FAR): percentage of incorrectly accepted impostors.

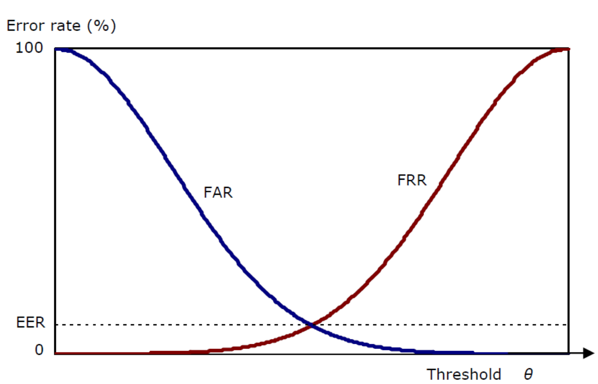

Fig. 2. False rejection rate and False acceptance rate as a function of the threshold θ.

Therefore, when designing a biometric verification system, the decision threshold (see Fig. 3) must be adjusted so that both errors are as low as possible, or one of the errors must be always below a certain threshold when a specific application requires this property.

A. Architecture of a speaker recognition system

A typical biometric recognition system consists of two phases (see Fig. 3): the training phase (enrollment) and the testing phase (recognition). In the training phase, biometric measurements from the users are captured by means of biometric sensors or readers. Then, relevant information is extracted from the biometric measurements (feature extraction) to build a user model, which will be stored in a database.

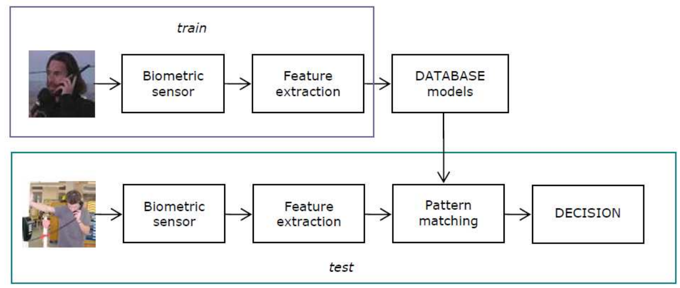

Fig. 3. Architecture of a typical biometric recognition system.

In the recognition phase, biometric readers are also used to capture biometric information of the user to be recognised. Relevant information is extracted from the data provided by the biometric sensors in the feature extraction step. This information is compared with the stored user models of the database, computing the degree of similarity (the term score is also used). This similarity measure will be used to determine whether the user corresponds to one of the users whose model is stored in the database or not. Finally, a decision is taken based on the computed similarity scores.

B. Feature extraction

Feature extraction or speech parameterisation in the speaker verification field consists in transforming the speech signal into a set of feature vectors [3]. The aim of this transformation is to obtain a relatively low-dimensional representation, more suitable for statistical modeling, the computation of a distance, or any other kind of score (in order to enable comparisons using simple similarity measures), while preserving the information related to the identity of the speaker.

The most commonly used parameters in state-of-the-art speaker and speech recognition technologies are the Mel-Frequency Cepstral Coefficients (MFCC) [5], [6]. They are a representation of the short-term power spectrum of a sound, based on the linear cosine transform of the log power spectrum on a nonlinear Mel scale of frequency (see Fig. 4).

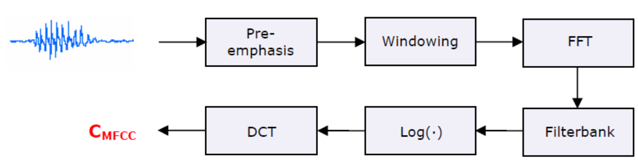

Fig. 4. Mel-frequency cepstral coefficients computation steps

Commonly, speech processing in the speaker verification task begins with a first order high-pass filtering of the speech signal to emphasise high frequency components. Then, the signal is segmented in temporal frames and typically windowed with a Hamming window to reduce the discontinuities in the boundaries of the segmentation. This procedure is usually used for short-term analysis of speech.

The first step for the computation of Mel-frequency cepstral coefficients is performing the Discrete Fourier Transform (DFT) of the speech frame. Usually a Fast Fourier Transform is used to reduce the computation time.

The resulting values are then passed through a filterbank distributed along the frequency domain according to a Mel scale. A vector of energy values is obtained with this step: the Filter Bank Energies (FBE). The Mel scale, proposed by Stevens [7] is based on the manner how the speech perception works in the human ear. The human auditory system non-linearly resolves frequencies across the audio spectrum. Empirical evidence suggests that a system that operates in a similar nonlinear way, obtaining the desired non-linear frequency resolution, provides a better recognition performance.

The Mel scale filterbank is a series of Q triangular bandpass filters that have been designed to simulate the bandpass filtering by mimicking the human auditory response. The series of constant bandwith triangular filters are 50% overlapped and spaced on a Mel frequency scale. On a linear frequency scale, the filter spacing is approximately linear in the range from 0 to 1000Hz, and logarithmic at higher values of frequency. The triangles are all normalised to have unit area.

After applying this filterbank, the number of coefficients is reduced, and hence the information is compacted. The variance is also reduced when averaging the samples of the DFT in each filter. Finally, a logarithmic compression and the Discrete Cosine Transform (DCT) is applied to the vector of FBE in order to obtain the MFCC.

The DCT serves two purposes. First, the DCT performs the final part of a cepstral transformation which separates the slowly varying spectral envelope (or vocal tract) information from the faster varying speech excitation. MFCC only retains the low order coefficients related to vocal tract.

The second purpose of the DCT is to decorrelate the elements of the feature vector. Elements of the log filterbank vector exhibit correlation due to both the spectral characteristics of speech and the overlapping nature of the filterbank. Such process makes the resulting decorrelated coefficients suitable for the use of diagonal covariance matrices in statistical classifiers.

Research on additional information sources in speaker recognition has been mainly focused on the use of the fundamental frequency as a complement to the vocal tract information provided by MFCC. One of the reasons is the robustness to acoustic degradations from channel and noise effects [8], [9]. Arcienega et al. [10], for example, suggest the use of F0-dependent speaker models. In the works of Sönmez et al. [11] and Adami et al. [12], the variation of fundamental frequency over time is modeled for its use in a speaker recognition task, together with the signal energy variation.

C. Statistical models

Gaussian mixture models (GMM) are commonly used as a modeling technique in speaker verification systems. A GMM is a weighted sum of gaussian density functions that models the distribution of the feature vectors extracted from the speech signal [13], [14]. Given a D-dimensional feature vector x, the Gaussian mixture model λ_i corresponding to the speaker S_i is defined by the expression in Equation 1.

$$P(x|\lambda_i) = \sum_{m=1}^{M} \omega_m N(x, \mu_m^i, \Sigma_m^i) \tag{1}$$

$N(x, \mu, \Sigma)$ is a gaussian function defined as shown in Equation 2, where μ is the vector of means and Σ the covariance matrix. D is the number of elements in the D-dimensional feature vector x. M is the number of mixtures, and ω_m are the weights of each mixture, that must sum up one.

$$N(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{2}$$

By means of dynamic programming algorithms, the probability that a sequence of speech frames was generated by this model can be determined [15], [16]. This probability -or likelihood- is used as a score for L frames of input speech given the model [15], [16], [4].

In the recognition step, given a sequence of test feature vectors $X = [x_1, x_2, ..., x_T]$ extracted from an unknown user's speech, the probability of the unknown speaker being the speaker S_i (assuming that vectors x_t are independent) is determined by the following expression in Equation 3, which will be used as a similarity score.

$$p(X|\lambda_i) = \prod_{t=1}^{T} P(x_t|\lambda_i) \tag{3}$$

D. Speaker verification system implementation

The speaker verification system implemented in this paper consists of state-of-the-art approaches. Two main speech features are used in the decision about the identity of the speaker: MFCC (13 coefficients calculated from windows with 256 samples, without window overlap) and fundamental frequency (F_0).

The proposed system performs cepstral mean substraction (CMS) to remove channel effects in MFCC parameters. Fundamental frequency and two derived parameters, relative jitter and relative shimmer, are also used in the likelihood computation of the speaker under analysis. Jitter is a measure of the periodic deviation in the voice signal, or the pitch perturbation of the signal. Shimmer (amplitude perturbation) is similar to jitter, but instead of looking at periodicity, it measures the difference in amplitude from cycle to cycle.

Frame	F0	JR	SR	LL
1	$F0_1$	JR_1	SR_1	LL_1
2	$F0_1$	JR_1	SR_1	LL_1
3	$F0_1$	JR_1	SR_1	LL_1
...	...	...	...	...
M	$F0_M$	JR_M	SR_M	LL_M
Mean	μ_{F0}	μ_{JR}	μ_{SR}	μ_{LL}

TABLE I
MEAN FEATURE VECTOR CALCULUS

Therefore, four parameters are extracted from each speech frame to evaluate the identity of the speaker: fundamental frequency, relative jitter, relative shimmer, and log-likelihood. The later is calculated as the difference between the log-likelihood of the feature vector (MFCC) for the claimed identity model, and the log-likelihood of the feature vector for the Universal Background Model (UBM). This UBM is the impostor model. Such model must contain every possible alternative to the speaker S_i.

In this paper was used the approach that consists in training the impostor model as a single model using several speakers [17], [18], [19]. This model is usually called universal background model (UBM), and when using Gaussian mixture models, the method is known as the GMM-UBM approach.

The similarity score between the unknown speaker and the claimed identity is performed comparing the mean vector of speech features (fundamental frequency (F0), relative jitter (JR), relative shimmer (SR), and log-likelihood (LL)) over all frames of the spoken utterance, as shown in Table I.

The similarity score between the input feature vector μ and the template S_i is given by a distance $d(\mu, \mu_{S_i})$. The distance measure between these two vectors can be expressed as $d(\mu, \mu_{S_i}) = (\mu - \mu_{S_i})\Sigma(\mu - \mu_{S_i})$. The Σ weighting matrix is the inverse covariance matrix corresponding to mean μ_{S_i}. This distance is known as the Mahalanobis distance. A threshold value θ, the maximum allowable value for distance $d(\mu, \mu_{S_i})$, is used to take the decision about the identity: known speaker or impostor.

III. Implementation of an Embedded Speaker Verification

The diagram of Fig. 5 shows the different components included in the embedded speaker verification system proposed in this paper.

The biometric sensor of the system is a microphone. The electric signal of the microphone is amplified and filtered to increase the Signal-To-Noise Ratio and to prevent aliasing during sampling.

The electric signal is sampled with the internal Analog-to-Digital Converter included in the microcontroller (dsPIC 33FJ128GP802-E/SP). The sampling rate is 16KHz and the resolution is 12 bits. The signal is segmented into frames of 256 samples, obtained using a double buffer DMA technique.

Each frame is processed to obtain different acoustic parameters, such as energy, spectrum, MFCC, fundamental frequency, jitter of fundamental frequency, and shimmer of amplitude values.

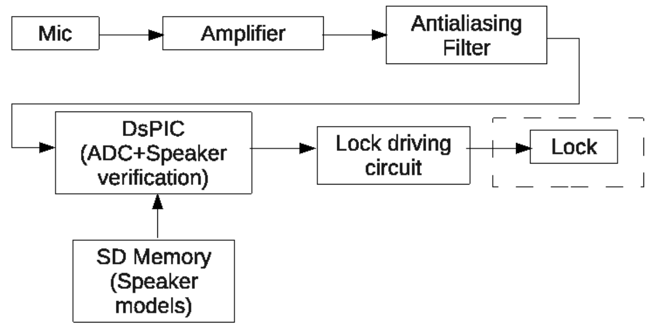

Fig. 5. Embedded speaker verification system.

These features are analyzed using the reference models, that are loaded into the dsPIC from an SD card. These models can be changed and reloaded into the SD card using the training software that creates a custom model for the specific user.

The system proposed in this paper was implemented using a device dsPIC 33FJ128GP802-E/SP from Microchip. The main features of this device are:

- 128KB of program memory. This makes it suitable for use with cross compilers.
- 16KB of RAM. Of which 2KB are shared with direct memory access (DMA) buffer as dual ported RAM.
- Up to 40 MIPS operation.
- Low cost. As mentioned before, its price is 4 US$, much lower than a classic DSP. This is a great advantage compared with comercial DSP devices.
- 16-bit wide data path.
- 12-bit@500ksps integrated analog-to-digital converter (ADC).
- Double-buffered input/output registers. This allows for faster operations on the ports (read-write), and also gives more flexibility on the handling of them.
- In-Circuit Serial Programing and Debbuger. The device can be programmed and configured in the end application circuit.
- 28-pin SOIC packaging available. Allows for great levels of integration.

A. Limitations in RAM memory of dsPIC

One of the main limitations to implement a speaker verification in dsPIC is the limited amount of RAM memory. In order to minimize the impact of such restriction, a number of aspects were taken into account to optimize the use of memory.

Without any optimization in the use of variables in RAM memory, the total amount of memory usage is shown in Table II.

Variables Buffer and Data are the integer and floating point representation of the frame under analysis. HammingWindow has the precalculated values of a 256 point Hamming window. The variables Sinus and Cosinus have the precalculated values

TABLE II
INITIAL MEMORY USAGE

Variable	#elements	type	Memory (bytes)
Buffer	256	int	512
Data	256	float	1024
HammingWindow	256	float	1024
Sinus	256	float	1024
Cosinus	256	float	1024
Butterfly	256	char	256
RealFFT	256	float	1024
ImagFFT	256	float	1024
MFCCFilters	24x53	float	5088
DCT	24x13	float	1248
Autocorr	256	float	1024
MFCCCoefs	13	float	52
InvCovars	13x2x16	float	3328
Means	13x2x16	float	3328
Priors	2x16	float	256
Determinants	2x16	float	256
Total			$21492 > 16$KB

TABLE III
ACTUAL MEMORY USAGE

Variable	#elements	type	Memory (bytes)
Buffer	256	int	512
Data	256	float	1024
HammingWindow	Program	memory	0
sigReal	256	long	1024
twdlFctr32b	768	long	3072
pwrspect/Autocorr	256	long	1024
MFCCFilters	Program	memory	0
DCT	Program	memory	0
MFCCCoefs	13	float	52
InvCovars	13x2x16	float	3328
Means	13x2x16	float	3328
Priors	2x16	float	256
Determinants	2x16	float	256
Total			$13876 < 16$KB

of a 256 point sinus and cosinus, and Butterfly has the necessary information for the correct calculation of the Fast Fourier Transform (FFT). The result of the FFT is put into the variables RealFFT and ImagFFT.

The Mel-frequency cepstral coefficients are calculated with the resulting spectrum of the FFT. The precalculated Mel scale filterbanks can be found in the MFCCFilters variable, and the final discrete cosine transform is computed using the DCT variable (precalculated cosinus).

Gaussian mixture models are loaded into the RAM memory of dsPIC from SD card into the variables InvCovars (inverse of the covariance matrices), Means (mean vectors), Priors (prior values of each gaussian mixture) and Determinants (determinant of each covariance matrix).

As shown in the last line, the total amount of memory necessary for our application is higher than the available memory.

A careful analysis of the variables shows that HammingWindow, MFCCFilters, DCT, and the routines that use them can be hard coded into Program memory. This decision saves 7360 bytes of memory that may be used for better Gaussian Mixture Models, that have a severe impact in the final performance of the speaker verification task.

The Fast Fourier Transform is now calculated using Microchip's subroutine FFTReal32bIP. This routine needs more memory, but the real benefit is obtained in speed.

The variable Autocorr is only used to calculate the fundamental frequency, which is then used to estimate jitter and shimmer parameters too. Therefore, Autocorr variable may point to the same memory space as pwrspect (Power Spectrum) variable , saving even more RAM memory. Actual memory usage of the system is shown in Table III.

B. Limitations in MIPS of dsPIC

Another important issue for the implementation of a embedded speaker verification system are the number of instruction per second executed in a dsPIC. The 40 MHz clock only allows 40 MIPS. Therefore, a careful selection in the implementation of the different routines is essential.

The original subroutine of Hamming window had this code:

```
int i;
for (i=0;i<bfrsize;i++)
{
 data[i]=data[i]*win[i];
}
```

The new subroutine of Hamming window has all the values hard-coded, with a benefit in memory usage and speed. The sample values of the Hamming window are literals (reduction in the usage of RAM memory), and the comparison and increment of for loop are not necesssary (increment in speed).

```
data[0]*=0.08f;
data[1]*=0.0801396318543067f;
data[2]*=0.0805584426474237f;
...
```

The same approach was used to implement the subroutines to calculate Mel-frequency cepstral coefficients, with a benefit RAM usage and speed. The original subroutines took 18.7ms to calculate the MFCC for a frame, while the new hard coded subroutine only takes 5.36ms.

The use of Microchip's subroutine for Fast Fourier Transform (FFTReal32bIP) has an important benefit in speed. The original subroutine coded in C had a duration of 64ms to obtain the power spectrum of a frame, while the new subroutine that includes the code proposed by Microchip only takes 5.36ms.

The integer implementation of autocorrelation was the final improvement in the code to process a frame and calculate all the necessary parameters to take a decision in the speaker verification task. The total time to process a frame is 59ms. The total time for the decision of the system depends on the duration of the utterance (without initial and ending silences). It may be estimated as 3.6 times the duration of the vocalization.

C. Speaker verification results

Several experiments were conducted using MATLAB and train-validation-test sets to study different algorithms for speaker verification systems, mainly focused in the optimization of the number of gaussian mixtures. GMM were trained using the standard Baum-Welch algorithm and diagonal covariance matrices.

The source code was written using both MATLAB and C30 sintaxis, and possible differences in the calculation of the algorithms was checked. The differences remain small and negligible.

Available speech from 168 speakers was divided into train and test sets. Both sets have different words, in order to build and test a text-independent speaker verification system.

The chosen architecture was tested using n-fold cross validation to obtain the ROC curve. The receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot of the sensitivity, or true positive rate versus false positive rate for a binary classifier system as its discrimination threshold is varied.

The ROC for the proposed system is shown in Fig. 6. As explained in Section II, when designing a biometric verification system, the decision threshold must be adjusted to minimize false positives and false negatives. In our task, false positives should remain below threshold to prevent an impostor acceptance.

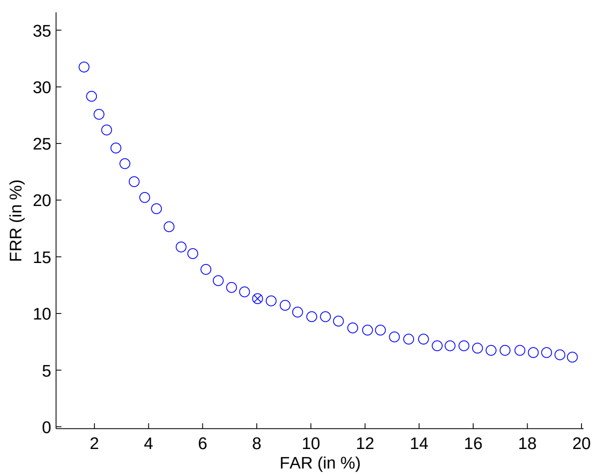

Fig. 6. ROC curve for the embedded speaker verification system.

IV. Conclusions

In this paper was described an embedded speaker verification system. The system is built in a low cost dsPIC from Microchip, which has characteristics of DSPs in a single 16-bit high-performance core. The speaker verification system is intended to operate an electrical door lock, through an electric lock driving circuit.

Experiments were performed using MATLAB to find the ROC curve of the proposed system, to get an approximation of the expected performance. Experimental results show that the system may reject impostor at the expense of also rejecting a user with the correct claimed identity. For example, the speaker verification system has a false acceptance rate of 8% for a false rejection rate of 12%. Actually, the proposed system can only be used as an auxiliary identification technique, and not as a primary identification technology, due to these low results.

Future work will focus in two main aspects: response time and better identification performance. Response time is an important issue in this 40 MIPS speaker verification system, and improvements can be achieved through the use of inline assembler inside C language.

Better identification performance will be faced with new speaker verification techniques, which may involved additional features, different modeling techniques, and signal conditioning algorithms.

References

[1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer, New York, 2003.

[2] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to Biometrics*. Springer, New York, 2004.

[3] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. Reynolds, "A tutorial on text-independent speaker verification," in *EURASIP Journal on Applied Signal Processing*, 2004, pp. 430–451.

[4] J. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437–1462, 1997.

[5] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, pp. 357–366, 1980.

[6] A. Oppenheim, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, pp. 95–99, 2004.

[7] S. Stevens, "The mel scale equates the magnitude of perceived differences in pitch at different frequencies," *Journal of the Acoustical Society of America*, vol. 8, pp. 185–190, 1937.

[8] B. Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustical Society of America*, vol. 52, pp. 1687–1697, 1972.

[9] M. Carey, E. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," in *Proceedings of the ICSLP*, 1996, pp. 1800–1803.

[10] M. Arcienega and A. Drygaljo, "Pitch-dependent gmms for text-independent speaker recognition systems," in *Proceedings of the Eurospeech*, 2001, pp. 2821–2825.

[11] K. Sönmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," in *Proceedings of the ICSLP*, 1998, pp. 3189–3192.

[12] A. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proceedings of the Eurospeech*, 2003, pp. 841–844.

[13] D. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91–108, 1995.

[14] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72–83, 1995.

[15] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, pp. 4–16, 1986.

[16] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[17] T. Matsui and S. Furui, "Likelihood normalization for speaker verification using a phoneme and speaker-independent model," *Speech Communication*, vol. 17, pp. 109–116, 1995.

[18] A. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *Proceedings of the ICASSP*, 1996, pp. 81–84.

[19] D. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proceedings of the Eurospeech*, 1997, pp. 963–966.