Digital Lock based on Speaker Recognition

Group D7

Sumedh Attarde (05D07012)
Joshua Abraham (05D07015)
Ritesh Kolte (05D07023)

Abstract

This report discusses the method and results of the implementation of a Digital Lock Based on Speaker Recognition. The Problem Statement involved using generic voice traits as a user discriminator. We have used the Linear Predictive Coding (LPC) to calculate Linear Predictive Cepstral Coefficients (LPCCs) to characterize the speech signal. We have implemented text-dependent speaker recognition i.e. recognition based on a fixed wordlist. If the 'distance' between the features of the trained words and the test words are above a similarity threshold the user is rejected.

1. Introduction

Speaker recognition, which can be classified into identification and verification, is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers. In this way, speaker recognition technology is expected to create new services that will make our daily lives more convenient. Another important application of speaker recognition technology is for forensic purposes.

Though the reliability of speaker recognition is less than fingerprint based verification or retina scan based verification, the results of the methods being currently researched are promising. In this project, we have attempted to build the digital lock using a processor as simple as possible.

There are a lot of methods in which speaker recognition can be implemented, but the basic software block diagram remains the same:

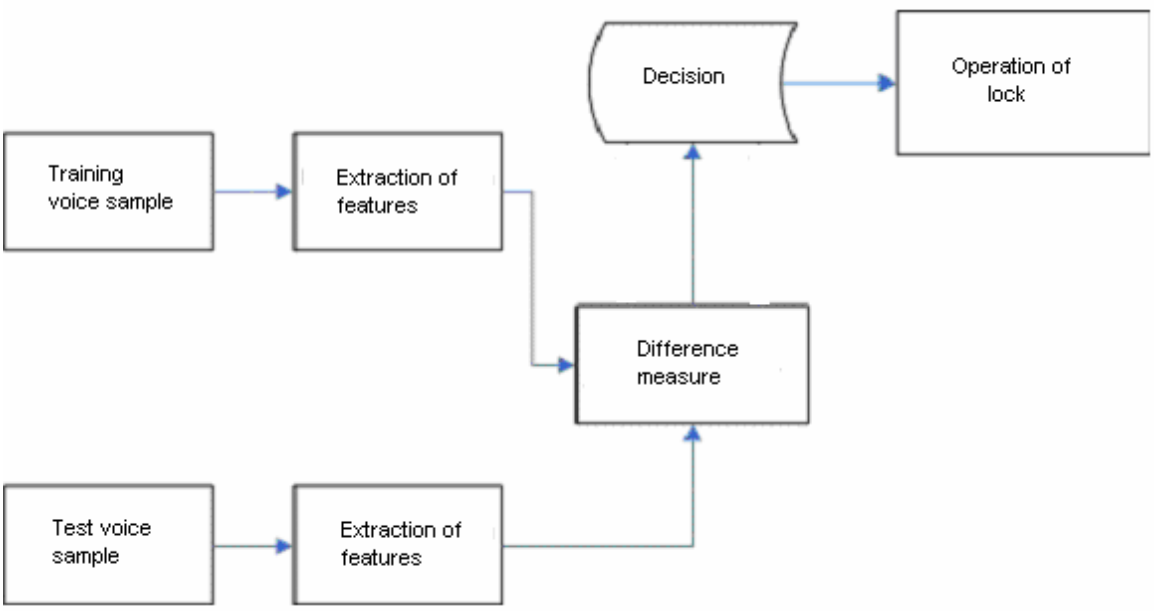


Fig. 1 Basic Methodology for Speaker Recognition

The input speech signal is obtained as a sequence of quantized values through an ADC. Different algorithms use different methods to extract features from the speech signal sequence. The feature elements are then compared with the stored template feature elements and the distance between them is computed. This distance, compared to a threshold, is used to make a decision whether the user is who he/she claims to be or an imposter.

2. Description of the software and hardware in this project

2.1 Software:

The feature elements used in this project are 10 Linear Prediction Cepstral Coefficients (LPCC) obtained from Linear Predictive Coding (LPC) coefficients for every 20 ms frame. The frames are the speech samples multiplied by Hamming window and have 50% overlap. The algorithm for computing the LPC coefficients, the Levinson-Durbin algorithm is described below:
Let x(n) be the speech sample sequence. The autocorrelation r(k) is defined as:

$$r(k) = \sum_{n=0}^{N-1-k} x(n)x(n+k)$$

Then the Mth order LPC coefficients are calculated as:

Initial values:

$E_0 = r(0)$

$a_{11} = \kappa_1 = r(1)/E_0$

$E_1 = E_0(1-\kappa_1^2)$.

With $m \geq 2$, the following recursion is performed

(i) $q_m = r(m) - \sum_{i=1}^{m-1} a_{i(m-1)} r(m-i)$

(ii) $\kappa_m = \dfrac{q_m}{E_{(m-1)}}$

(iii) $a_{mm} = \kappa_m$

(iv) $a_{im} = a_{i(m-1)} - \kappa_m a_{(m-i)(m-1)}$ for $i = 1, \cdots, m-1$

(v) $E_m = E_{m-1}[1-\kappa_m^2]$.

(vi) If $m < M$, then increase m to $m+1$ and go to (i). If $m = M$, then stop.

Given the LP coefficients $\{a[k]\}_{k=1}^{p}$, cepstral coefficients $c[n]$ are computed using the following recursive formula

$$c[n] = \begin{cases} a[n] + \sum_{k=1}^{n-1} \frac{k}{n} c[k]a[n-k], & 1 \leq n \leq p \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c[k]a[n-k], & n > p. \end{cases}$$

We extracted 10 LPCC from the 10 LPC per frame and stored them in the feature matrix. When a user claims to be a particular person, the feature matrix for his voice signal is obtained and compared with the stored template feature matrix of the claimed person using a technique known as the Dynamic Time Warping, described below:

The distances between every feature vector of the template sequence of vectors from every feature vector in the test sequence of vectors is calculated and stored in a matrix (old_matrix) of dimensions (length of template sequence)x(length of test sequence). A new matrix of the same dimensions is constructed whose elements are formed from the earlier matrix using the rule:
new_matrix(i, j) = old_matrix(i, j) +
minimum (new_matrix(i, j-1), new_matrix(i-1,j), new_matrix(i-1, j-1))

The element new_matrix (trained sequence length, test sequence length) gives the distance between the test sequence and template sequence. Various kinds of optimizations can be used to reduce the number of computations; different weightage to different elements can be included in the above equation to be more effective for the particular task at hand.

The distance measure calculated using the above method is matched with a threshold. If the calculated distance is above the threshold, the user is regarded as an imposter and the lock fails to open, otherwise a decision in favour of the person's claim is made and the lock opens.

2.2 Hardware:

The hardware comprises five main parts. The analog front end that processes the condenser microphone input; the MSP430F247 processor; the user interface(LCD and DPDT switches); the external memory (SRAM); and the motor based lock.

- The analog front end part comprises of an 8-pole 4-stage stage Butterworth low pass filter cum post-amplifier that performs anti-aliasing and denoising functions. This also limits the microphone signal amplitudes and provides the ADC with signal levels that it can decode at the same time while maximizing the signal levels.

- The MSP430F247 processor, though not meant for heavy computations which are required for speech processing, is sufficient if we restrict the number of users and time is not too much of a constraint.

- The external SRAM was needed for any appreciable speech processing, as the internal memory of the MSP was far from sufficient for the purpose. The features from the voice samples are computed and stored in the SRAM.

- The user-interface is the LCD and the keypad. The user keys in his information through the keypad, which is simultaneously displayed on the LCD. The processor issues appropriate directives to the user through the LCD, when necessary.

- The motor based lock moves based on the microcontroller decision. If the user is deemed to be an imposter, the lock won't budge. Otherwise, the lock opens, stays open for a while and closes. The L293D motor driver is used to link the MSP to the motor.

3. Reasons for the choices of methods and hardware:

3.1 Software:

There are a variety of features currently being used for speaker recognition such as Mel Frequency Cepstral Coefficients (MFCC), different features derived

from Linear Predictive Coding etc. or other methods based on Hidden Markov Models (HMM), Gaussian Mixture models (GMM) or even neural networks. Further, to reduce the size of the feature elements being stored, clustering techniques such as Vector Quantization (VQ) are used.

Considering the limited memory and computational power of the MSP, we had to adopt both a computationally less intensive method which provides reasonable accuracy and a method which doesn't require large temporary memory. The LPCC were chosen because they don't involve operations such as the FFT, operations in the math library, unlike the MFCC. But the temporary memory was still much beyond the capabilities of the MSP (See 3.2). The factor which we were ready to compromise upon was the time required to come to the decision, but we didn't suffer heavily in this issue. Also, the fixed point operation resulted in further compromise in the accuracy, as we had to do scaling to eliminate the risk of overflows.

Effects: We had to restrict the spoken word time to 1 second. Also, we had to resort to calculating feature vectors for all frames AFTER the entire signal was obtained because the MSP wasn't able to finish off the required computations by the time the ADC gave the next sample.

3.2 Hardware:

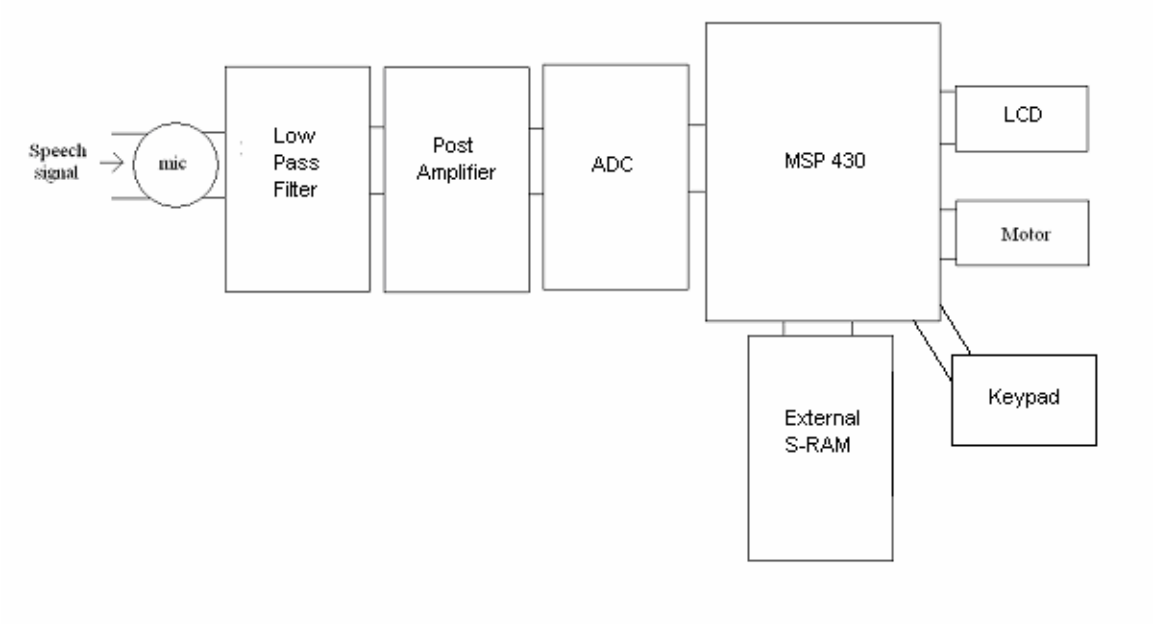
The memory of the MSP was very limited for the purpose at hand. Hence external SRAM was used. All computations had to be done by reading some values at a time from the ex-SRAM, processing and writing the output at the appropriate addresses in the ex-SRAM. The MSP was chosen as it seemed a significant improvement over the ATMEGA16 we used last semester for the project. It is a 16 MHz, 16 bit processor with a 12 bit internal ADC. Standard speech processors use 24 bit data but for our purposes a DSP chip seemed too much.

Since the application involved transfer of large amounts of data, we used a parallel access SRAM for high speed access. Peak memory usage involved storage of two 100x100 matrices of type long int, along with two vectors of 10 elements each (type long int). Analysis of these requirements resulted in use of 1Mbit SRAM organized as words of 16bits.

The Analog front end was changed from last semester. Distinctive voice characteristics are known to occur even in the 4-10 KHz frequency range and hence we decided to sample at 20 KHz to improve the accuracy of the recognition. In order to improve dynamic range of the ADC sampling, a post-amplifier was

included. The L293D is a bipolar driver. We needed to open and close the lock and needed bi directional movement capability of the motor.

4. Block diagram



5. Development of the project and the problems faced

We were faced with the effects of fixed point computations, which introduced severe overflow risks. To avoid the overflows, we used the Q24.8 representation and defined new multiply and divide functions which scaled the output appropriately. This multiplication and division took more time than the plain multiply and divide. These newly defined functions were tested for accuracy exhaustively by checking for the expected output using numerous inputs, which covered all possible cases.

6. Discussion of the results

Currently we are facing a problem with the microprocessor. The timer used for starting the ADC is not receiving its source clock (SMCLK).

7. Conclusions and Future Work

We have to iron out the problem related to sourcing of the timer clock..

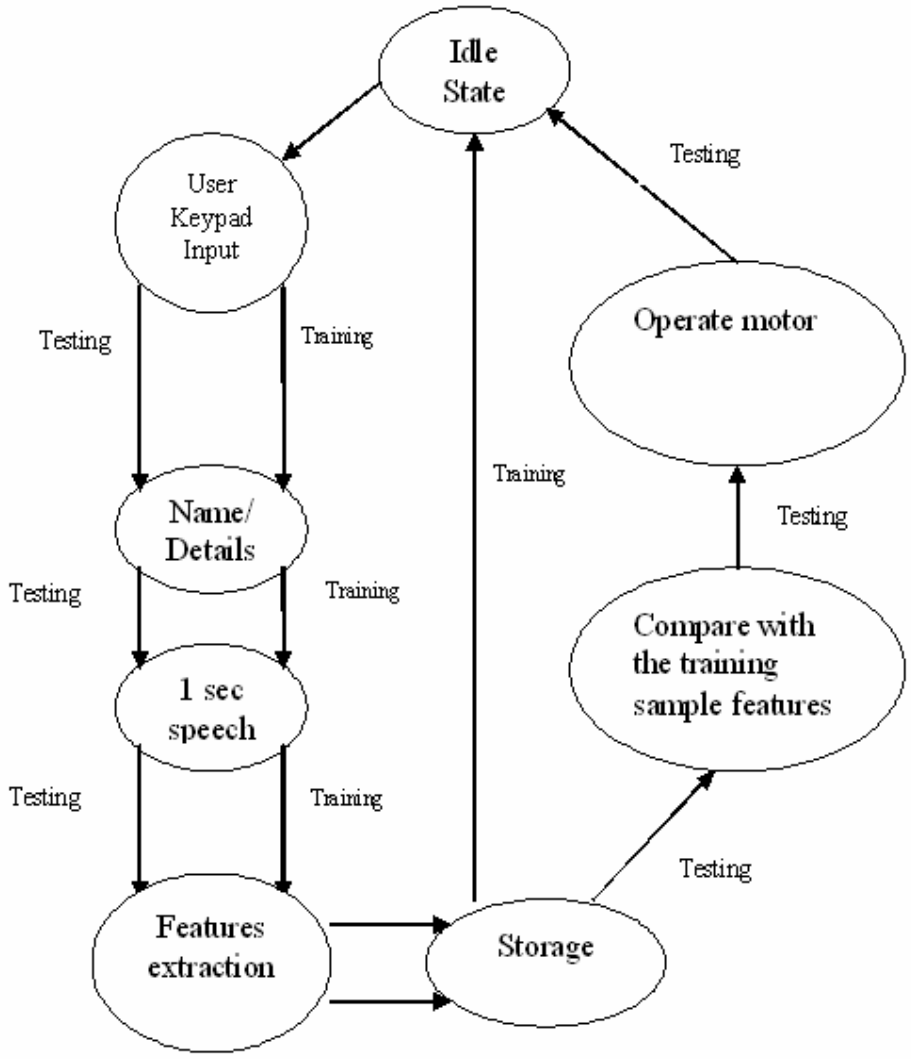
8. References

[1] The MSP430F2xxx datasheet and MSP430x2xx User Guide provided by TI.
[2] "Linear Predictive Speech coding" by Mr. Xavier Serra, Music Technology Group, Dept. of Information and Communication Technologies & Audiovisual Institute, Universitat Pompeu Fabra, Barcelona. (His homepage was last updated on 24/11/2008.)
[3] "Dynamic time warping": http://en.wikipedia.org/wiki/Dynamic_time_warping
[4] Fixed point math tutorial put up on a forum: http://www.gamedev.net/community/forums/topic.asp?topic_id=214292
[5] "Spectral Features for Automatic Text-Independent Speaker Recognition" by Tomi Kinnunen, University of Joensuu. (Ph.Lic. thesis)

9. Appendix

Keypad inputs:

TRAIN DOWN UP RIGHT SELECT


Idle State
User Keypad Input
Testing
Training
Name/ Details
Testing
Training
1 sec speech
Testing
Training
Features extraction
Storage
Training
Testing
Operate motor
Testing
Compare with the training sample features
Testing