Design of Digital Blowing Detector

Hessa Al-Junaid, Amina Mohamed Saif, and Fatima Yacoop AlWazzan

***Abstract*—In this paper, a Digital Blowing Detection (DBD) algorithm is developed to identify human's blow sound signal despite variation in sounds signals. Also this project aims to advance the use of the blowing as input mechanism in controlling application. The algorithm was developed by investigating the sound signals characteristics using digital processing techniques and then utilizes the acquired results to detect the blow sound signal. The proposed method used is an elimination method based on the unique energy levels distribution of voice and unvoice signals. A microphone and MATLAB environment were used to develop the algorithm (capture, record and process microphone input signals). The results of the experiments indicate, that the DBD algorithm successfully eliminate non-blow signals and detecting the blow signals with a 75.7%. This paper demonstrated that human blow is a suitable input to devices which operate based on ON-OFF command, also when a rapid response is required, this include a mouse click with no additional hardware is required for the implementation.**

***Index Terms*—Human computer interaction, non-speech sound detection, non-verbal vocal input, blow detection, assistive technologies.**

I. Introduction

Human body is considered as a rich source for input, where voices, face, fingers, hands, and eyes have all been explored and employed in different applications in Human-Computer Interface HCI. Many alternative input mechanisms are being developed such as haptic input devices, eye trackers, foot operated devices, emotions, gestures, and speech recognition. Speech recognition systems focus on the verbal aspects of the voice, where the system accepts verbal speech as input and convert it to words and sentences and it takes some time for the system to recognize it. The use of speech recognition effectively depends on many factors such as, type of microphone, its position to speaker, background noise, software used, pronunciation, speaking rate, vocal characteristics.

On the other hand, paralanguage using non-speech (nonverbal) sounds such as blowing, humming, whistling, or hissing as inputs of information in HCI has minor investigation in the research community. Non-verbal sounds are characterized by properties such as pitch, volume, loudness, vowel quality, continuation, and timbre. Their variations are controlled by the human. There are various applications in the literature based on those properties such as mouse cursor [1], [2], joystick [3], multimedia [4], games, and keyboard emulation [5]. For instance, Sporka [1] used non-speech sounds by means of control by pitch of the tone to emulate the mouse and keyboard of a PC to be used by motor impaired users. He found that some non-speech sounds can dominate the movement better than speech sounds. Igarashi and Hughes [6] presented several techniques using the non-verbal features in the voice to enhance traditional voice recognition systems. One of the presented techniques' controls by Continuous Voice' describes the idea of using the voice as on/off button. Further, they used other techniques such as rate-based parameter control by pitch and discrete parameter control by tonguing to control one dimensional joystick.

Another system is the Vocal Joystick (VJ), which has been developed by a group of computer scientists at the University of Washington [3]. The system allows users with motor impairments to use their voice characteristics and vowel sounds to control an on-screen mouse pointer, robotic arm, wheelchair, normal joystick signals. Their interface exploits a large set of continuous acoustic phonetic parameters like pitch, loudness, vowel quality.

Although the use of vowels as an input is a novel technique, it introduces cognition-related latency due to the difficulty of remembering the vowel that corresponds to each direction. Al-Hashimi [4] overcomes the Vocal Joystick latency weaknesses by replacing the vowel sounds with the breathing (blowing) sound in developing the Blowtter. Blowtter is a voiced controller plotter that allows a disabled user to blow into a mic-board in order to move the head of the plotter to draw. Assistive technology highlighted the importance of switching action for motor impaired people. Yvonne [7] suggested a phoneme recognition to offer another advantage over speech recognition in that it may provide a method of controlling a continuously varying parameter through varying the length of the phoneme or the pitch of a vowel sound. Temporal and spectral features that characterize different phonemes are explored to enable phoneme distinction. The system was implemented both in hardware and software.

There are many benefits of using blowing and other paralinguistic vocal control, this include simplicity of the design compared to the possibility to maintain a near-real-time and continuous response, the potential of cross-cultural use, the relatively low cost and the technical ease of using the microphone as an input device, and the possibility of use by people with motor impairments especially if their impairments are accompanied by speech impairments. From the summarized survey, the non-verbal voice has many advantages such as they are simple design, language independent, accurate in noisy environment, used for immediate response which is near real time. Non-verbal voice used in interface or process can be used in variety of applications.

Manuscript received November 13, 2015, revised March 13, 2016.

The authors are with the Department of Computer Engineering, College of Information Technology, University of Bahrain, Kingdom of Bahrain (e-mail: haljunaid@uob.edu.bh).

doi: 10.18178/ijiee.2016.6.3.620

II. Methodology

A set of experiments are done to record human blow. Matlab is used to facilitate recording various sound waves using a microphone with noise cancelation functionality [8]. Since our target is to design a blowing detector that detects any blowing sound, and not to train the model to detect the blowing of an individual. It was required to study different sounds and resolve a way to identify blow sound despite a variation in sounds. The method we decided to use is Elimination Process, which is a method to identify an entity of interest among several ones by excluding all other entities. Elimination process is the core of this work methodology, where the main process is to exclude the blowing sound leaving out all other sound types. In order to achieve this result, an identification method is required to identify the features and parameters which characterize the blowing sound. The method steps are explained in this section.

A. Obtaining and Processing Samples

1) Materials and methods

In order to resolve a way to identify the blowing sound from other sounds, a noise canceling microphone connected to laptop with installed Matlab software is used to record different sound samples and process them in both time-domain and frequency-domain. Recording sound directly in Matlab on a computer with attached or integrated microphone requires you to specify the number of samples to record, the sample rate, number of channels and sample format [9].

In the experiments, the sounds of different speakers are used to record different samples in different locations. Most samples have been recorded under normal room condition, with no noise interference. Because the samples were recorded to study the difference between blowing and other sounds, different silence, voiced and voiceless samples has been recorded. For voiceless samples, the focus was on the voiceless fricative ignoring the stops and affricative because the first group is continues sounds as well as the blow, while the other groups are instantaneous voiceless. The voiceless fricative used for the test samples are /f/, /th/, /s/, /sh/, and /h/ (The initial sounds of "fin," "thin," "sin," "shin," and "him.") [10]. Using Matlab recording program record.m, 200 samples sound files were saved in WAVE format at 16 kHz sampling rate, mono channel and 2 second length in the following manner: 30 silence samples, 50 blow samples (male and female), 20 speech samples (male and female), and 20 samples for each one of the 5 fricatives.

2) Observations and results

Initial analysis in time and frequency domain was done to identify the characteristic of silence, speech, and blow signals.

The analysis illustrated the fact that silence signals are non periodic with insignificant amplitude, while the spectrogram reflects a low energy for low frequencies. With respect to speech signals, they are periodic and the affricative sounds are characterized by the presence of formants in the spectrogram.

Formants as seen in Fig. 1 for /n/, /z/, in speech signals are resulted because of vibration in the vocal cords. Fig. 1 shows the spectrogram of sample fricatives, affricative, and blow signals. The blow and voiceless fricatives do not have formants because the vocal cords do not vibrate in their condition. Also, voiceless fricatives and blow signals are non-periodic. Further, they have high energy levels at almost all frequency bands 0-7kHz and this is due to the fact that they face less resistance at vocal tract than the other voiceless sounds. Characteristics of each fricative was identified in spectrogram with respect to the distribution of energy.

Based on the above analysis to differentiate the blow signals from both speech and voiceless fricative signals, the blow detection algorithm was developed. It was found that the periodicity and the formant lines are the key elements to eliminate the speech sounds (voiced). After that an additional elimination for the silence and voiceless fricative is required in order to obtain the blow signal. The latest elimination will be based on the different energy-frequency distribution for each signal type.

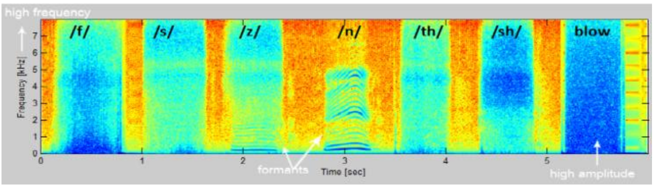


Fig. 1. Spectrogram of fricatives, affricative, and blow signals.

III. Blow Detection Algorithm

Sound Signals have slowly varying nature, therefore, it is common to process sound in frames (or blocks). Over which the properties of the sound waveform can be assumed to remain relatively constant over very short intervals (5-20ms). Sound signal is processed frame-by-frame in overlapping intervals until entire region of the signal is covered, this process is illustrated in Fig. 2. In this work, the input signal is 16 kHz, then the typical value for the frame length is 320 (20ms) with 50 % overlap.

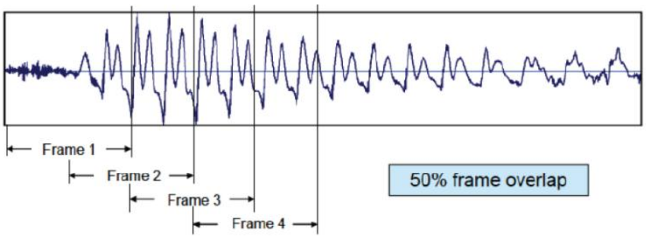


Fig. 2. Signal frame-by-frame processing.

After obtaining the frame block, the next step is to apply a window to each frame in order to reduce signal discontinuity at either end of the block [11]. The hamming window is a commonly used window and it is calculated as in Eq. 1, where k is the windows length

$$\omega(k) = 0.54 - 0.46\cos(\frac{2\pi k}{k-1}) \qquad (1)$$

A. Voiced-Unvoiced (Voiceless) Decision

There are a variety of approaches that can be used in the classification of voiced speech, unvoiced speech, and silence, which are: the energy of the signal, zero-crossing rate of the signal, autocorrelation, LPC first predictor coefficient, and

LPC energy [12]. The five measurements parameters can be used individually or in combination to make the decision.

For this work, the short-time autocorrelation parameters is used to reveal periodicity in a signal, because it is considered robust indicator of periodicity. For a given discrete signal, the short-time autocorrelation function is generally defined as in Eq. 2:

$$R(m) = \frac{1}{N}\sum_{n=0}^{N-1-m} x(n).x(n+m), 0 \le m < M_0 \quad (2)$$

where N is the length of the analyzed sequence, M_0 is the number of autocorrelation points to be computed, and m is called lag or delay. Essentially, the signal $x(n)$ is being convolved with a time-lagged version of itself. The autocorrelation function has an important property which is periodicity. The function has a one major limitation that it can retain much information present in the signal. In speech, many peaks present in the autocorrelation function are due to damped oscillations of the vocal tract response. Center clipping as a spectrum flatteners technique, appears the best to make the periodicity more prominent while suppressing other features which may cause distracting peaks. Center-clipping technique is used for $x(n)$ in a pre-processing stage of the decision algorithm, to give a center-clipped signal $y(n)$ based on Eq. 3, where C_L is the clipping level (threshold).

$$y(n) = \begin{pmatrix} x(n) - C_L & x(n) \ge C_L \\ 0 & |x(n)| < C_L \\ x(n) + C_L & x(n) \le -C_L \end{pmatrix} \quad (3)$$

The received sound frame block is first filtered using a low-pass Butterworth filter. Then, a certain percentage of the waveform is clipped using center clipping method. Using MATLAB build-in function xcorr, the clipped data autocorrelation is found. After obtaining the autocorrelation sequence, the sequence is searched for a maximum peak Amax in the lags between 20-100 samples (corresponding to periodicity of speech). If the maximum peak is above third of the correlation value at lag=0, then the frame was considered voiced. If not, the frame is identified as unvoiced.

B. Blowing Decision

The last stage in the algorithm is the decision if a voiceless frame is a blow signal or not. To achieve this, and as an outcome from the last section, it is required to compare different signal energy-frequency distribution rates. Other than blow signals, unvoiced signal may be either silence or voiceless fricatives. Examining those signals we noticed that by dividing the signals into four bands with respect to their frequency component which is presented at the spectrogram y-axis as illustrated in Fig. 3. Signals can be distinguished by comparing the average energy level at each band.

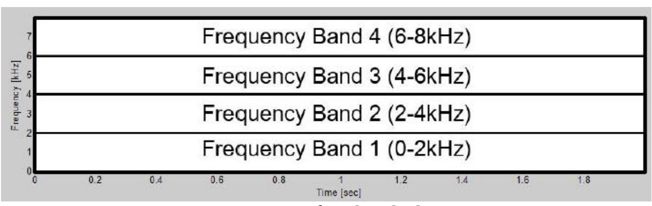


Fig. 3. Frequency four bands division.

Voiceless signal samples frequency bands division are illustrated in Table IV. Energy distribution for each type is unique and the identification will be based on this finding. The blowing decision algorithm is broken into a four steps, calculate the covariance Power Spectral Density (PSD), then divide it to four bands of frequencies, average the energy at each band, and then compare the bands to decide on the blow signal.

1) Calculating the covariance PSD

The power spectral density (PSD) or energy spectral density is a positive real function of a frequency variable associated with energy. It is often called simply the spectrum of the signal. PSD is one of three spectrum spectral estimators, the other two are mean square spectrum (MSS) and pseudo spectrum. MATLAB has different estimators for each of these three types. To calculate the PSD one of 7 different estimators supported by MATLAB need to be created and passed to either psd or psdopts function. In this work, covariance estimator cov was selected and send it to the psd function in order to obtain the PSD data. It returns a PSD object with the spectrum computed as a function of physical frequency (Hz). Fs the sampling frequency specified in Hz, the object Y is the distribution of power per unit frequency. For real signals, PSD returns the one-sided PSD by default. Note that a one-sided PSD contains the total power of the input signal.

2) Divide PSD data into frequency bands with average energy

Since the previously calculated PSD object Y is the distribution of power per unit frequency, the next step (step 2 and step 3 combined) will be dividing the object data into four bands and calculate the average energy at each bands. The length of the PSD data is found to be 257. Then, each band length will be approximately 63-64. The average energy is calculated using the mean function which returns the mean value of the elements of an array, see listing 1.

```
AvgPSD = [0 0 0 0] ; %4 Bands Average PSD
for j = 1 : 4
  if ( j ==1) %Band 1
   AvgPSD( j ) = mean (Y. d a t a ( 1 : 6 4 ) ) ;
     e l s e i f ( j ==2) %Band 2
   AvgPSD( j ) = mean (Y. d a t a ( 6 5 : 1 2 8 ) ) ;
     e l s e i f ( j ==3) %Band 3
   AvgPSD( j ) = mean (Y. d a t a ( 1 2 9 : 1 9 2 ) ) ;
    e l s e %Band 4
   AvgPSD( j ) = mean (Y. d a t a ( 1 9 3 : 2 5 7 ) ) ;
  end
end
}
```

Listing 1. Matlab code of Voiced-Unvoiced Decision.

The AvgPSD vector holds the average PSD energy of each frequency band, this means that AvgPSD(1) is Band1 average energy, AvgPSD(2) is Band2 average energy, and so on.

3) If-selection blowing decision

The algorithm compares the calculated average energy at each band, with pre-defined threshold calculated for the blow signal in order to eliminate non-blow signals. Earlier examples demonstrated in Fig. 4, show the average energy distribution in the frequency bands for each voiceless sound. The same concept was applied for all test samples we recorded. Examining the the results, If selection decision was written in three fundamental steps as illustrated in the flow chart in Fig. 6.

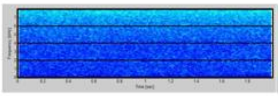
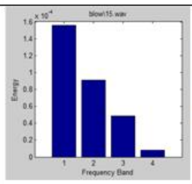
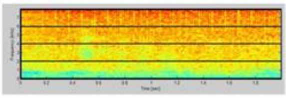
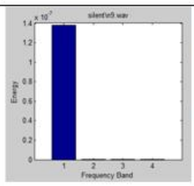
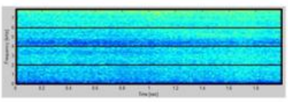
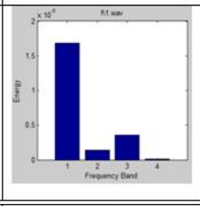
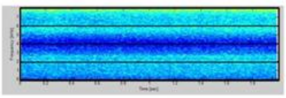
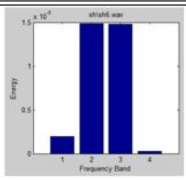
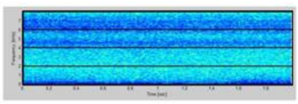
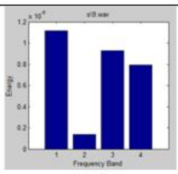
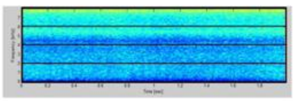
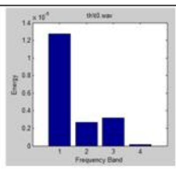
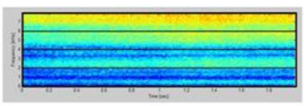
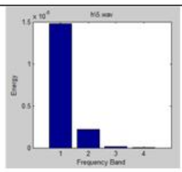
Signal	Band division	Average Energy Level at each Band	
Blow			1.50e-4 9.09e-5 4.88e-5 8.30e-6
silence			1.38e-7 6.20e-11 4.00e-12 4.80e-12
/f/			1.68e-5 1.39e-6 3.60e-6 1.55e-7
/sh/			1.94e-6 1.48e-5 1.48e-5 2.89e-5
/s/			1.11e-6 1.35e-7 9.30e-7 7.94e-7
/th/			1.27e-5 2.66e-6 3.16e-6 1.62e-7
/h/			1.48e-5 2.18e-6 1.72e-7 6.61e-9

Fig. 4. Voiceless signals band division.

4) Silence elimination

Compared to the blow signal, silence signal has low energy mainly in frequency Band-4. Silence signal Band-4 energy average is either multiplied by 10^{-11} or 10^{-12}, while blow signal Band-4 energy average is usually multiplied by 10^{-6}. Therefore, silence signal will be eliminated by checking the value of Band-4 energy. This step is represented by the blue decision block in the flowchart in Fig. 6.

5) Fricative /s/ and /sh/ elimination

Fig. 5 illustrates the three signals blow, /s/, and /sh/. It is apparent that they have different energy density at different bands. Band-3 for /sh/ signal contains the highest energy average, Band-2 for /s/ contains the lowest energy average, and the blow signal has a consistent decreasing shape form the highest Band-1 to the lowest Band-4. This varying in the averages value was dedicated to eliminate both the /s/ and /sh/ signals. The flowchart Fig. 6 shows this operation with 3 pink decision blocks.

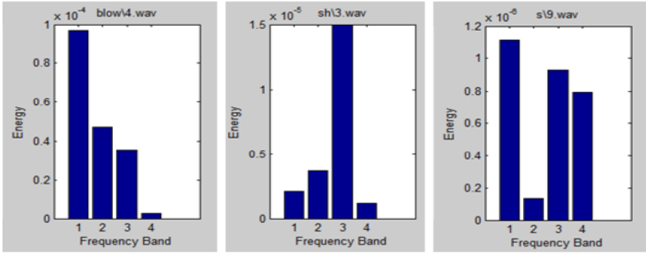


Fig. 5. Blow, /sh/ and /s/ frequency bands energy.

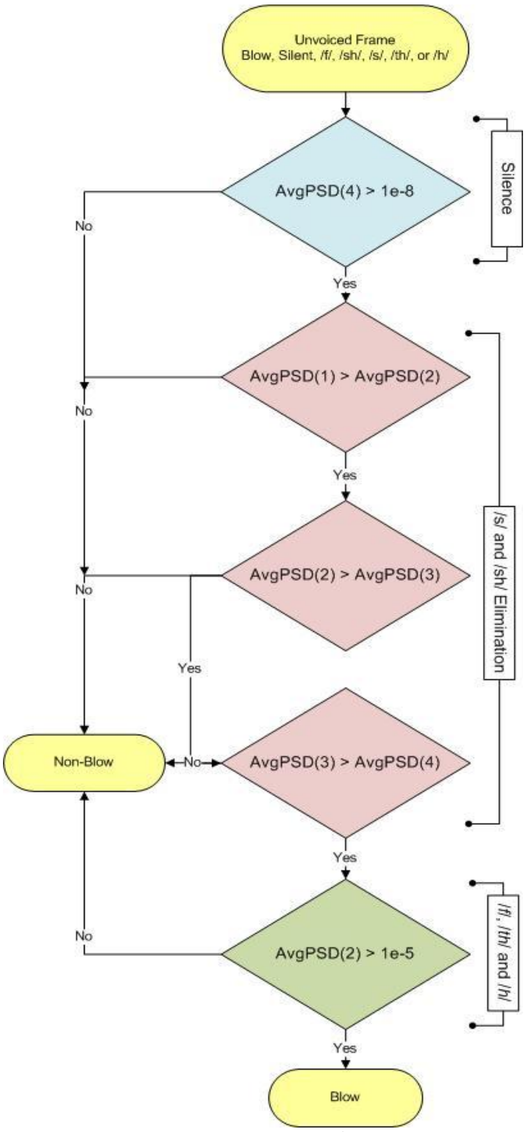


Fig. 6. Elimination process flowchart.

6) Fricative /f/, /th/ and /h/ elimination

This step is similar to the silence elimination step, where we are going to eliminate all remaining fricatives by examining the average energy value of Band-2 and compare it to a threshold. The threshold value is said to be 10^5, since all remaining fricative have a lower energy at Band-2. The green decision block in the flowchart reflects this step Fig. 6. The elimination processes is carried by a set of If-command in Matlab as shown in Listing 2.

```
i f (AvgPSD( 4 ) > 1e-8)
    i f (AvgPSD( 1 ) > AvgPSD ( 2 ) )
        i f (AvgPSD(2)>AvgPSD ( 3 ) )
            i f (AvgPSD(3)>AvgPSD ( 4 ) )
              i f (AvgPSD(2)>1e-5)
               %Blow Frame
        end
      end
    end
  end
 end
}
```

Listing 2. Matlab code for Elimination Process.

After completing this process, which identify if an input frame is blowing or not, a broader decision-making has to be done. The algorithm has to decide whether the entire signal (wave file input) is a blowing signal or not. A counter blow has been defined to count the total number of blowing frame in the signal, if blow is greater than half the nframe (total number of frame) the signal is considered blow signal, otherwise the input is non-blow signal.

IV. Observations and Results

The Blowing Detection Algorithm has been tested using 200 samples. The algorithm successfully identified 28 blow samples out of 37, which means that 75.7% of the blowing samples were recognized. The BDB algorithm successfully identified the blow signal samples by excluding all other signals type. The DBD algorithm shows 0% error rate in excluding non-blow signals, whereas 24.3% error rate was presented in identifying the blow signals. For the time being, the error rate percentage is acceptable, future modification to the algorithm selection criteria can be done in order to reduce the error amount. Compared to existing blow utilize applications, applications using our algorithm has the advantage of being entirely dedicated to blow signal inputs. Blow Windmill and Blow Balloon are mobile entertainments applications (apps) that we tested and found that the application process is activated with any high amplitude utterance indifferent whether it is blow or not. On the other hands, Al Hashimi Blowtters [4] blow detection algorithm was frequency-based where high pitch signals are considered as blow, implies that Blowtter can be activated using the /s/ utterance.

The DBD algorithm can be utilized in the following suggested applications, 1) Switching any device ON and OFF; 2) Control TV settings Blow TV remote control; 3) Switching to next slide during a presentation; 4) Gaming and Entertainment. This work makes an initial step towards the use of non-speech input mechanisms in the area of human-computer interface. During the execution of the project, some interesting ideas have been set out in order to continue with the research. The DBD algorithm selection criteria are only simple approach to detect the blow. More future experimentation is needed in order to come to accurate conclusions. During this study, a MATLAB code has been written to control the Mouse events using blowing as well as switching the presentation slide in Microsoft PowerPoint.

V. Conclusion

A Blowing Detector algorithm was developed based on an elimination method. The elimination method is based on the energy levels distribution of signals over 4 bands. The algorithm filters out speech, fricative, silence signals based on identification of their unique energy distribution. This work makes an initial step towards the use of non-speech input mechanisms in the area of human-computer interface.

References

[1] A. Sporka, "Non-speech sounds for user interface control," PhD thesis, Czech Technical University, Prague, 2008.

[2] S. Kaur, "Mouse movement using speech and non-speech characteristics of human voice," *International Journal of Engineering and Advanced Technology*, vol. 1, June 2012.

[3] J. Bilmes *et al.*, "The vocal joystick: A voice-based human-computer interface for individuals with motor impairments," in *Proc. Human Language Technology Conference on Empirical Methods in Natural Language Processing*, October 2005.

[4] S. Al-Hashimi, "Paralinguistic vocal control of interactive media," PhD thesis, Lansdown Center for Electronic Arts, Middlesex University, Hertfordshire, United Kingdom, 2007.

[5] A. Sporka, S. Kurniawan, and P. Slavk, "Non-speech operate emulation of keyboard," in Cambridge Workshop on Universal Access and Assistive Technology, Designing Accessible Technology, 2006.

[6] T. Igarashi and J. Hughes, "Voice as sound: Using non-verbal voice input for interactive control," *UIST*, Orlando, Florida, ACM, November 11-14, 2001.

[7] Y. M. Nolan, "Control and communication for physically disabled people, based on vestigial signals from the body," PhD thesis, National University of Ireland, Dublin, 2005.

[8] J. Hansen. Record program. (2001). [Online]. Available: http://www.ele.uri.edu/hansenj/projects/ record/record.pdf

[9] I. Mcloughlin, *Applied Speech and Audio Processing With MATLAB Examples*, Cambridge University Press, 2009.

[10] H. Fu, R. Rodman, D. McAllister, D. Bitzer, and B. Xu, "Classification of voiceless fricatives through spectral moments," MS thesis, 1999.

[11] N. Mikael and E. Marcus, "Speech recognition using hidden markov model," Department of Telecommunications and Speech Processing, Blekinge Institute of Technology, 2002.

[12] L. Rabiner, "Speech recognition based on pattern recognition approaches," *Digital Speech Processing*, vol. 155, pp. 111–126, 1992.

Hessa Al-Junaid is an assistant professor of computer engineering at the University of Bahrain, College of Information Technology. She received the B.Sc. and M.Sc. degrees in electronics engineering from the University of Bahrain, Bahrain, in 1996 and 2001, respectively. In 2006, Dr. Al-Junaid received her Ph.D. degree from the University of Southampton, Southampton United Kingdom. Dr. Al-Junaid was a chairperson for Computer Engineering Department from 2010 to 2013. Her research interests include modeling and simulation of mixed-signal circuits and systems and mixed-disciplinary systems, brainwaves modeling and analysis for brain machine interface and robotics applications. Dr. Al-Junaid is an active member in the Institution of Engineering and Technology IET Bahrain chapter.

Amina Mohamed Saif is a computer system administrator at Bahrain Polytechnic. She received the B.Sc. degree from the University of Bahrain, Sukhair, Bahrain, in 2012 in computer engineering. She is currently working toward the M.Sc. degree in information technology at the same university. Her research interest includes digital signal processing, audio signal processing, communication, audio compression and encoding.

Fatima Yacoop AlWazzan is a computer teacher at the Ministry of Education in Bahrain. She received the B.Sc. degree in computer engineering from University of Bahrain, Bahrain, in 2012. In 2014, AlWazzan received the apprenticeship diploma degree in education from Ministry of Education, Bahrain. She is currently working toward the B.Sc. degree in apprenticeship in education from Ministry of Education, Bahrain.