Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Neural Network Society, Vol. 8, Iss. 3, pp. 148-156, 2010.

A SPOKEN WORD BOUNDARIES DETECTION STRATEGY FOR VOICE COMMAND RECOGNITION

IGOR S. PERETTA, GERSON F. M. LIMA, JOSIMEIRE A. TAVARES, KEIJI YAMANAKA

Computer Engineering Department, Electrical Engineering Faculty,
Federal University of Uberlandia – P.O. Box 593, 38400-902, Uberlandia, MG, BRAZIL

E-mails: iperetta@gmail.com, gersonlima@ieee.org, josycbelo@gmail.com, keiji@ufu.br

Abstract — The use of voice commands as a new way of interaction between man and machine is the subject of several researches in recent years and has already been produced commercial and freeware applications. However, considering the achieved results, there is still a great development potential in this area, particularly in Brazilian Portuguese language. This work proposes: 1. an efficient method of detecting spoken word boundaries from a recorded signal, using Teager Energy Operator and FIR Filter; 2. the use of wavelet transform and wavelet packet filter bank as a main tool for feature extraction to feed a multi-layer artificial neural network to recognize a limited vocabulary of voice commands. The system was developed using a dataset of spoken words from 50 speakers, using normal pronunciation speed and in an environment without any noise control. Tests with the system show a very good classification rate and noise robustness.

Keywords — Voice command recognition, spoken word boundaries detection, teager energy operator, discrete wavelet transform, wavelet packet filter bank, artificial neural network.

1. Introduction

The speech recognition and voice command recognition are an extensive area of research with many possible applications in our daily lives, while they could simplify many everyday tasks, enable new forms of human-computer interaction, generate innovative controls to the development of expanded reality, or even support the inclusion of disabled people with severe restrictions of movement. However, after many years of world-wide researches, there is not still an ultimate application.

Some factors prevent the complete success of this objective. They could be listed: the indeterminacy of equipment’s quality that will be used to capture the voice; the different levels of noise to which applications are always subjected; the inherent differences of each independent speaker; even considering the same speaker, there will be speech changes in different situations, caused by illness, fatigue, or even the so-called Lombard effect; the lack of understanding of all human hearing biological and cognitive processes. Particularly, Brazilian Portuguese is a greater challenge for speech recognition, considering the amazing variety of accents throughout the Brazilian territory.

Several conventional preprocessing techniques are well known to speech recognition, as extraction of LPC coefficients [1], [2], or Mel-frequency Cepstrum coefficients [3]. Wavelets are also used in speech recognition field [4], [5]. Comparison between some of them can be found in literature [6], [7].

This research has two main premises: first, artificial neural networks (ANN) have achieved several successes in speech patterns recognition; second, sound analysis made by human ears can be represented by wavelet transforms, at least in its first stage which is determined by the response function of the human cochlea [8]. The use of wavelet functions to increase robustness to noise has also been shown, by emulating frequency resolution of the human cochlea [4].

The proposed system was implemented for recognition of a limited vocabulary in Brazilian Portuguese with six voice commands: “SOBE” (up), “DESCE” (down), “AZUL” (blue), “VERMELHO” (red), “DIREITA” (right), and “ESQUERDA” (left). The diagram of the system is shown in (Fig. 1).

The used database, obtained with the freeware Audacity® [9], contains three recording versions of each of the six voice commands. They are voices of 50 speakers, 30 males and 20 females aged between 17 and 40 years. Thus, the database has a total of 900 samples in the Waveform audio format (WAV). The Audacity® software was configured with a sampling frequency of 8kHz and a length of 16 bits per sample of signal amplitude. The recordings were made using a simple computer microphone, with 75Ω of impedance, in a room with a steady stream of people and no control of noise.

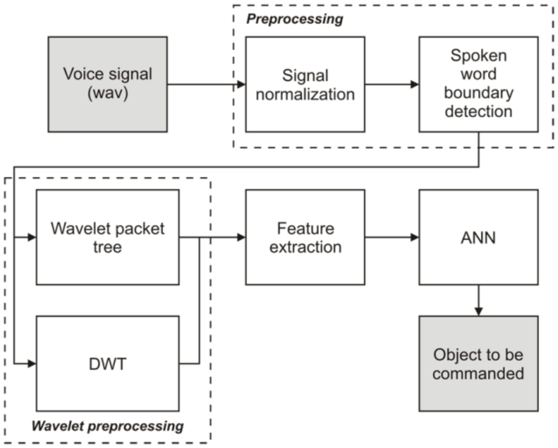


Figure 1. Application functional diagram

Results related in this paper improve previous work results [10] and a comparison between them is made in subsection 3.1. Introducing the Teager Energy Operator (TEO) [11] and implementing of a Lowpass FIR Filter with Blackman windowing [12], the boundaries detection algorithm has gotten real improvements. A comparison between TEO + FIR method and conventional methods are presented in subsection 2.1.1. Thus, the ANN could be trained with more reliable patterns and it has increased its capacity for recognition and generalization [13].

This paper reports the achieved improvement obtained in the voice command recognition project conducted by the Computer Engineering Department, in Federal University of Uberlandia (UFU). New results are presented in this work, and compared with Mel-frequency Cepstrum and “Bottom-up” methods in subsection 3.2.

2. Project Development

2.1 Preprocessing

This is one of most important stage in any signal processing application and the improvements achieved in the results were obtained by restructuring the preprocessing algorithms.

First, signal recordings (sampled signals) are normalized and proportionally transformed to maximize the fit into the amplitude spectrum defined by the interval between -1 and 1. The main idea is to minimize the impact of different amplitudes in speakers’ voice signals. Note that noise and interference amplitudes present in the sampled signal are also affected.

The next stage is to detect the beginning and the final points of the spoken word (possible voice command) within each sampled signal, i.e., detecting the spoken word boundaries. To reach this goal, we start by using TEO operator, defined (in continuous or discrete domain) as “very useful 'tools' for analyzing single component signals from an energy point-of-view” [11]

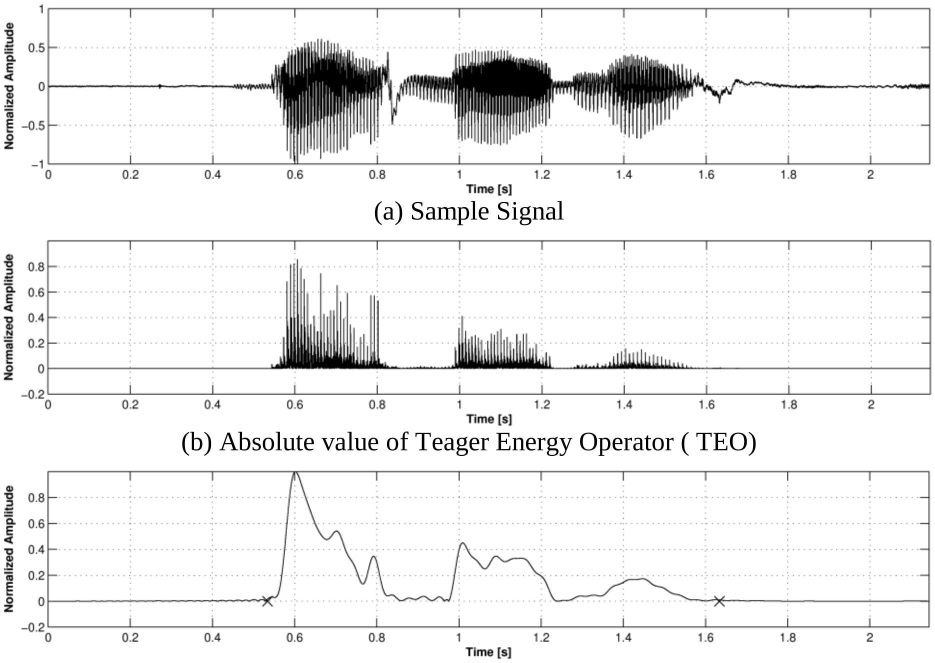
The equation (1) presents TEO´s definition in the discrete domain:

$$\Psi[x(n)] = x_n^2 - x_{n-1} \cdot x_{n+1} \qquad (1)$$

where *x* is the analyzed vector (signal)

In Fig. 2.a and 2.b, is presented the absolute values of the TEO applied to the sampled signal. This representation of the signal minimizes the noise present in silent periods of the speaker. As can be seen in Fig. 2.b, the resulting signal after application of the TEO operator emphasizes the region of the spoken word syllables. Fig. 2.c shows the signal result after the FIR Filter. The used lowpass FIR filter were designed with a transition bandwidth of a 0.1 rad/s, cutoff frequency of 0.125 Hz, and a Blackman windowing of 440 samples.

An algorithm that runs through the signal envelope from the beginning of the window, detects the starting point of the spoken word (a possible command). Similarly, from the end of the voice signal, the algorithm detects the end of the spoken word (see crosses on Fig. 2.c). In this way, each original sampled signal is cropped according to its respective detected boundaries and the resulting signal is referred in this paper as the “spoken word signal”.



(a) Sample Signal

(b) Absolute value of Teager Energy Operator (TEO)

(c) Envelope based on TEO / [×] Boundaries of a possible voice command

Figure 2. Analysis of a “VERMELHO” signal recording (Portuguese for “red”)

2.1.1 Comparison of TEO+ FIR with Classical and “Bottom-up” Methods

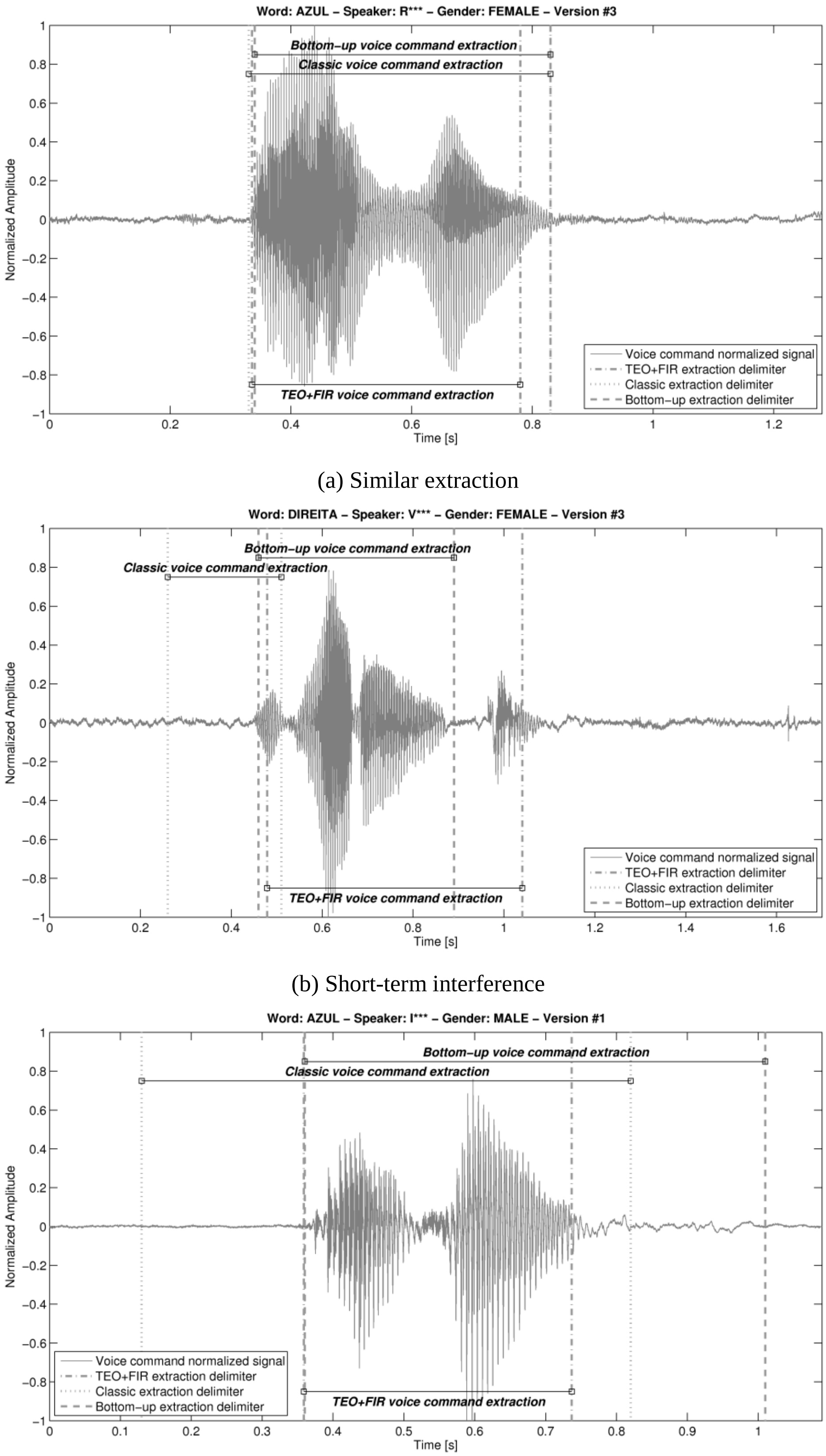
To provide a comparison basis for the word extraction method using TEO and FIR filter, two other word extraction methods were implemented: a Classical method that uses Energy and Zero-Crossing rate computations [14], and the “Bottom-Up” method, proposed by Lamel, Rabiner, et al [15]. All methods were applied to the 900 samples database used in this work. Fig. 3 presents four relevant sample results.

Fig. 3.a shows the case where there is basically white Gaussian noise in the environment. In this case, all methods found similar boundaries to word extraction, i.e., both extracted words are understandable to human hearing.

Fig. 3.b presents the case where there is short-time interference in the signal, before the voice command. Classical method tends to capture this interference, and the TEO + FIR method ignores the interference. TEO + FIR method tends to capture the region of highest energy inside the signal. It is possible to observe that if the interference is not short, TEO + FIR method will also include the interference when dealing with boundaries decision. In the case of “Bottom-Up” method, it worked almost like TEO + FIR method, but it missed the last delayed syllable.

In certain noisy environments, Classical method tends to miss the correct word extraction boundaries, as presented in Fig. 3.c. The zero-crossing rates of noisy environments are significantly higher than when the white noise is considered. If there is the presence of non-white noise, “Bottom-Up” method can´t find boundaries correctly, as can be seen after the end of word signal. TEO + FIR method is more robust.

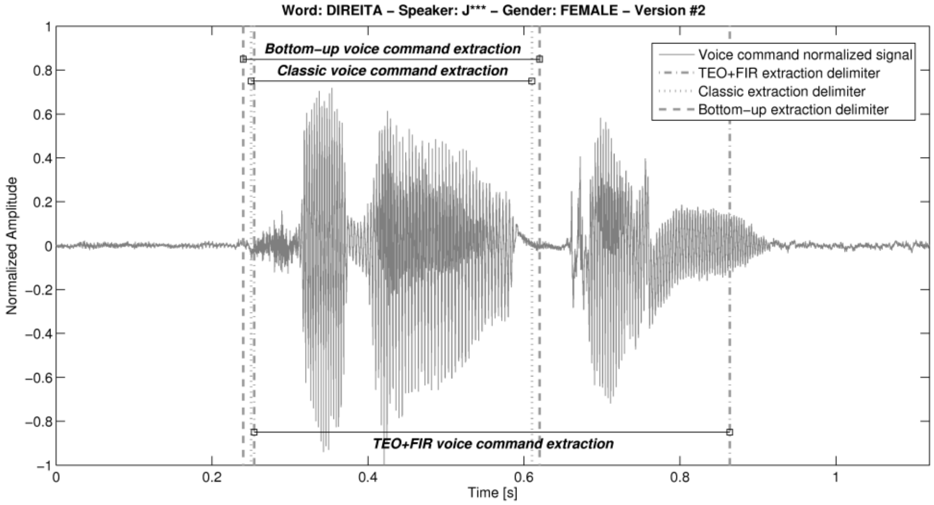
The proposed method based on TEO+FIR works forwards and backwards over the signal to detect the start and endpoint boundaries. Fig. 3.d presents the case where the speaker delayed his speech. The Classical method tends to set the endpoint before the delayed syllables, similar to what was made by “Bottom-up” method.



(a) Similar extraction

(b) Short-term interference

(c) Non-white noise



(d) Delayed syllables

Figure 3. Comparisons of proposed boundaries detection method and classical method

2.2 Wavelet Preprocessing

There are two independent processes used for each signal. The first one corresponds to a **wavelet packet decomposition** of the signal, generating a binary tree with the format presented in Fig. 4. Coefficients of each tree node correspond to the response in time domain when we pass a signal through different frequency Bandpass filters. The frequency bands shown in table 1 were designed to simulate the human hearing system [4]. The other process concerns the application of a **1-D discrete wavelet transform (DWT)** to decompose the signal. Both decompositions are at level 5 with mother-wavelet "db4" (Daubechies 4). Results of this stage include **coefficients** from the **wavelet tree nodes** and from **DWT** used to analyze the spoken word signal.

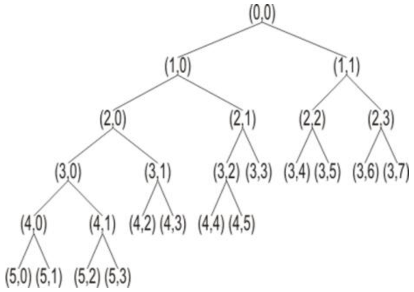


Figure 4. Wavelet packet tree

Table 1. Frequency bands at each wavelet packet tree node.

Frequency bands [Hz]	*Wavelet packet tree node*
0-125	(5,0)
125-250	(5,1)
250-375	(5,2)
375-500	(5,3)
500-750	(4,2)
750-1000	(4,3)
1000-1250	(4,4)
1250-1500	(4,5)
1500-2000	(3,3)
2000-2500	(3,4)
2500-3000	(3,5)
3000-3500	(3,6)
3500-4000	(3,7)

2.3 Feature Extraction

Based on information generated in earlier stages, four sets of features are brought to us:

- Number of concentration regions on spoken word signal.
- Correlated-energy at each node from the spoken word signal wavelet tree.
- Entropy from each pre-determined interval of the spoken word signal.
- Correlated-intensity from intervals of spoken word signal DWT decomposition.

Number of concentration regions in a spoken word signal: To calculate this number of concentration regions, the implemented algorithm uses the curve of standard deviation of the signal to emphasize the areas already highlighted by TEO's envelope signal. In speeches with controlled pronunciation, it is not hard to intuit that the number of concentration regions of the signal is the amount of syllables in the word pronounced. In this experiment, there was no control of sampled independent speaker's pronunciation, so, this information was not verified. The quantity of concentration regions is the first element of the features vector.

Correlated-energy at each node from the wavelet tree: Coefficients of wavelet packet tree were the basis for calculating the energy in each node of interest. As each speaker has different ways to pronounce the same word, we chose to correlate the energy values instead of storing their calculated values. This set creates 13 new elements to the features vector, one for each tree node.

Entropy from each pre-determined interval of the spoken word signal: The spoken word signal is divided into 16 identical intervals to calculate the entropy of each one. We considered Shannon entropy. Entropy values found are 16 new elements to the features vector.

Correlated-intensity from intervals of DWT decomposition: The algorithm takes **DWT** coefficients found earlier and separated them in six groups, each one with its proper meaning and size (approximation coefficients at level 5, and detail coefficients at level 5, 4, 3, 2, 1) as shown in Fig. 5. Table 2 shows for each group how many intervals were predetermined. At each interval, its intensity is calculated by equation (2).

$$I(n_1 : n_2) = \frac{E}{\Delta t \cdot A} = \frac{\sum_{i=n_1}^{n_2} x_i^2}{(n_2 - n_1 + 1) \cdot \sum_{j=n_1}^{n_2} \left| x_j \right|} \qquad (2)$$

where n_1, n_2 are boundaries of interval Δt of the signal x, E is the interval energy, and A is its area.

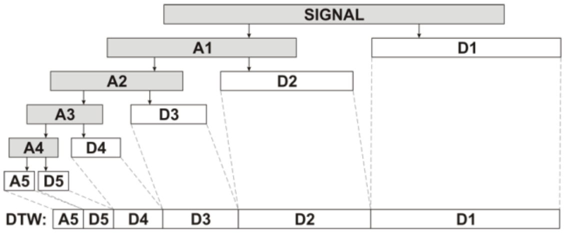


Figure 5. DWT signal decomposition.

Table 2. Set intervals to calculate intensities.

Set of signal decomposition	*Proportion to the total size of the signal*	*Number of intervals*
D1	2^{-1}	64
D2	2^{-2}	32
D3	2^{-3}	16
D4	2^{-4}	8
D5	2^{-5}	4
A5	2^{-5}	4

As speakers has different ways to pronounce the same word (utterance) and the noise from the environment can influence the signal intensity, this motivated to also correlate the intensity values, instead of storing their calculated ones. They made 128 more elements to be added to the features vector.

After finalizing this stage, **158 features** are extracted from each spoken word signal. Features vectors of all analyzed signal recordings form the features arrays that, after being normalized, are used to train the ANN.

2.4 Artificial Neural Network

To recognize voice commands, an ANN with multi-layer architecture (MLP) was implemented due to its high ability to recognize new patterns (robustness).

A single hidden layer MLP was implemented with 158 input units, 100 hidden units and 6 output neurons. Each output neuron corresponds to a voice command and its activation is equivalent to recognition. The hyperbolic tangent function was chosen for activation of the neurons. ANN was configured with bipolar inputs and binary targets.

Different algorithms were tested to verify a fast training for ANN. *Scaled Conjugate Gradient* [16] was chosen as supervised learning algorithm and *Bayesian Regularization* (also known as *Mean Squared Error with Regularization*) [17] was chosen as performance evaluation algorithm.

From the total of 900 recorded patterns from 50 speakers, the ANN was trained with:

- Number of Patterns for Training: 600 (utterance 1 and 2 of each speaker)
- Number of Patterns for Testing: 300 (utterance 3 of each speaker)
- Max number of Epochs: 1000

3. Results and Conclusion

Confusion matrices are typically used in pattern recognition applications as a visualization technique. Each row data of a confusion matrix represents the ANN recognized output class, while each column represents the desired target class. This type of matrices enables the system confusion analysis, presenting the application performance.

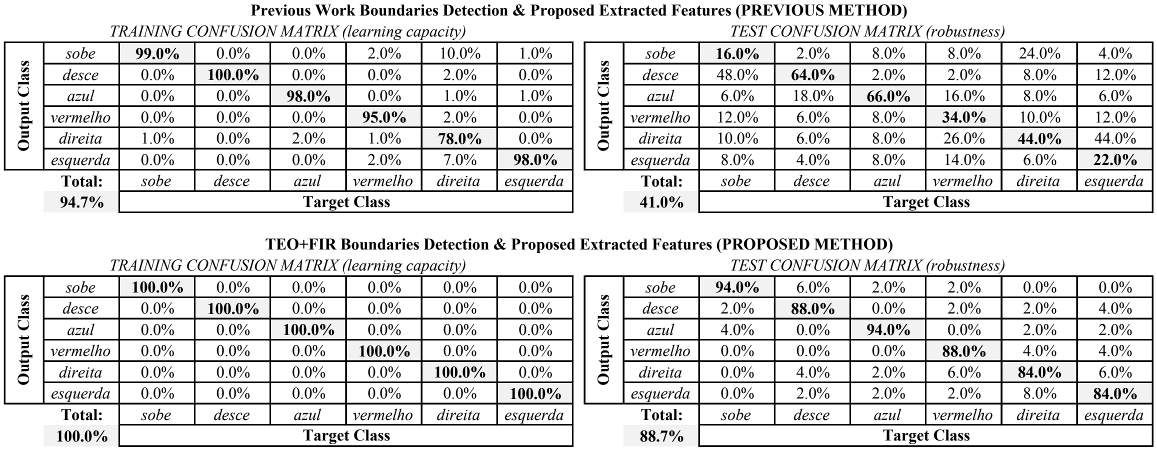
In the following subsections, confusion matrices are adopted to enable the comparison of the proposed methodology with the previous work [10] methodology and with a conventional methodology based on Mel-frequency Cepstrum coefficients.

3.1. Comparison with Previous Work

The main difference between previous and present work is the preprocessing described in subsection 2.1. Previous preprocessing stage uses an algorithm based on original signal envelope to detect word boundaries. Previous work extracts the same features as described in subsection 2.3.

The TEO+FIR boundaries detection algorithm and the proposed feature extraction presented in this paper are part of a methodology here named *proposed method*. In order to compare with previous work results, a rerun of its obtained extracted features was necessary with the actual ANN architecture. Table 4 presents results for comparison, naming previous work methodology as *previous method*.

Table 4. Confusion matrices for comparison with previous work



Previous Work Boundaries Detection & Proposed Extracted Features (PREVIOUS METHOD)

TRAINING CONFUSION MATRIX (learning capacity)

Output Class	*sobe*	**99.0%**	0.0%	0.0%	2.0%	10.0%	1.0%
	desce	0.0%	**100.0%**	0.0%	0.0%	2.0%	0.0%
	azul	0.0%	0.0%	**98.0%**	0.0%	1.0%	1.0%
	vermelho	0.0%	0.0%	0.0%	**95.0%**	2.0%	0.0%
	direita	1.0%	0.0%	2.0%	1.0%	**78.0%**	0.0%
	esquerda	0.0%	0.0%	0.0%	2.0%	7.0%	**98.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	94.7%	**Target Class**					

TEST CONFUSION MATRIX (robustness)

Output Class	*sobe*	**16.0%**	2.0%	8.0%	8.0%	24.0%	4.0%
	desce	48.0%	**64.0%**	2.0%	2.0%	8.0%	12.0%
	azul	6.0%	18.0%	**66.0%**	16.0%	8.0%	6.0%
	vermelho	12.0%	6.0%	8.0%	**34.0%**	10.0%	12.0%
	direita	10.0%	6.0%	8.0%	26.0%	**44.0%**	44.0%
	esquerda	8.0%	4.0%	8.0%	14.0%	6.0%	**22.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	41.0%	**Target Class**					

TEO+FIR Boundaries Detection & Proposed Extracted Features (PROPOSED METHOD)

TRAINING CONFUSION MATRIX (learning capacity)

Output Class	*sobe*	**100.0%**	0.0%	0.0%	0.0%	0.0%	0.0%
	desce	0.0%	**100.0%**	0.0%	0.0%	0.0%	0.0%
	azul	0.0%	0.0%	**100.0%**	0.0%	0.0%	0.0%
	vermelho	0.0%	0.0%	0.0%	**100.0%**	0.0%	0.0%
	direita	0.0%	0.0%	0.0%	0.0%	**100.0%**	0.0%
	esquerda	0.0%	0.0%	0.0%	0.0%	0.0%	**100.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	100.0%	**Target Class**					

TEST CONFUSION MATRIX (robustness)

Output Class	*sobe*	**94.0%**	6.0%	2.0%	2.0%	0.0%	0.0%
	desce	2.0%	**88.0%**	0.0%	2.0%	2.0%	4.0%
	azul	4.0%	0.0%	**94.0%**	0.0%	2.0%	2.0%
	vermelho	0.0%	0.0%	0.0%	**88.0%**	4.0%	4.0%
	direita	0.0%	4.0%	2.0%	6.0%	**84.0%**	6.0%
	esquerda	0.0%	2.0%	2.0%	2.0%	8.0%	**84.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	88.7%	**Target Class**					

Table 4 shows a better recognition performance of the proposed method. The improvement of robustness for the trained ANN is strongly verified. While the previous method reaches 41.0% of total correct recognition rate for the test set, the proposed method reaches a total of 88.7%. The minimum improvement ratio of 1.38 can be verified on correct recognition rates of "DESCE" patterns, while "SOBE" patterns present the improvement ratio of 5.88 on correct recognition rates (almost an improvement of 6 times).

The results improvement obtained here when compared to previous work [10] are directly related to the improvements made at the preprocessing stage, in particular, the proposed method of detecting spoken word boundaries into a recorded signal, using operator TEO and implemented FIR filter.

3.2. Comparison with a Conventional Methodology

To evaluate this proposed method, a conventional preprocessing method was chosen from a PhD thesis [6] which evaluates several techniques. The better combination for voice command recognition using ANN found in this thesis is the use of later named "Bottom-Up" algorithm for word boundaries detection [15] and Mel-frequency Cepstrum coefficients as inputs for the ANN. In this work, this combination is identified as *BMC method*.

BMC method uses 20ms windows and variable size frames to extract 1280 Mel-frequency Cepstrum coefficients from the segmented voice command, independently of its size. This coefficients extraction algorithm was applied to our database of voice commands to generate the inputs of the ANN. Table 5 presents confusion matrices generated by proposed method and BMC method.

Table 5. Confusion matrices for comparison with BMC method

TEO+FIR Boundaries Detection & Proposed Extracted Features (PROPOSED METHOD)

TRAINING CONFUSION MATRIX (learning capacity)

Output Class	*sobe*	**100.0%**	0.0%	0.0%	0.0%	0.0%	0.0%
	desce	0.0%	**100.0%**	0.0%	0.0%	0.0%	0.0%
	azul	0.0%	0.0%	**100.0%**	0.0%	0.0%	0.0%
	vermelho	0.0%	0.0%	0.0%	**100.0%**	0.0%	0.0%
	direita	0.0%	0.0%	0.0%	0.0%	**100.0%**	0.0%
	esquerda	0.0%	0.0%	0.0%	0.0%	0.0%	**100.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	100.0%	**Target Class**					

TEST CONFUSION MATRIX (robustness)

Output Class	*sobe*	**94.0%**	6.0%	2.0%	2.0%	0.0%	0.0%
	desce	2.0%	**88.0%**	0.0%	2.0%	2.0%	4.0%
	azul	4.0%	0.0%	**94.0%**	0.0%	2.0%	2.0%
	vermelho	0.0%	0.0%	0.0%	**88.0%**	4.0%	4.0%
	direita	0.0%	4.0%	2.0%	6.0%	**84.0%**	6.0%
	esquerda	0.0%	2.0%	2.0%	2.0%	8.0%	**84.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	88.7%	**Target Class**					

"Bottom-Up" Boundaries Detection & Mel-frequency Cepstrum Coefficients (BMC METHOD)

TRAINING CONFUSION MATRIX (learning capacity)

Output Class	*sobe*	**99.0%**	0.0%	0.0%	1.0%	0.0%	0.0%
	desce	0.0%	**100.0%**	0.0%	0.0%	0.0%	0.0%
	azul	0.0%	0.0%	**100.0%**	0.0%	0.0%	0.0%
	vermelho	0.0%	0.0%	0.0%	**99.0%**	1.0%	0.0%
	direita	1.0%	0.0%	0.0%	0.0%	**98.0%**	0.0%
	esquerda	0.0%	0.0%	0.0%	0.0%	1.0%	**100.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	99.3%	**Target Class**					

TEST CONFUSION MATRIX (robustness)

Output Class	*sobe*	**88.0%**	6.0%	10.0%	0.0%	4.0%	0.0%
	desce	2.0%	**92.0%**	0.0%	2.0%	2.0%	6.0%
	azul	6.0%	0.0%	**86.0%**	2.0%	0.0%	0.0%
	vermelho	0.0%	2.0%	2.0%	**88.0%**	4.0%	12.0%
	direita	0.0%	0.0%	0.0%	2.0%	**86.0%**	8.0%
	esquerda	4.0%	0.0%	2.0%	6.0%	4.0%	**74.0%**
	Total:	*sobe*	*desce*	*azul*	*vermelho*	*direita*	*esquerda*
	85.7%	**Target Class**					

Table 5 shows a slightly better recognition performance of the proposed method. However, some relevant differences could be identified between results from both methods:

- The proposed method is more robust when it presents 84.0% as a minimum local recognition rate as can be seen for "DIREITA" or "ESQUERDA" of the test set, while BMC method minimum recognition rate is 74.0% for "ESQUERDA".
- The proposed method is more accurate when it presents a maximum of 94.0% recognition rate, as can be seen for "SOBE" or "AZUL" of the test set, while BMC method maximum recognition rate is 92.0% for "DESCE".
- The proposed method presents a better balance between correct recognition rates for the test set, with a standard deviation of 4.50%, while BMC method presents a standard deviation for correct recognition rates of 6.12%.
- The BMC method presents a higher recognition error rate. For example, it misrecognizes 12.0% of "ESQUERDA" patterns as "VERMELHO", and 10.0% of "AZUL" patterns as "SOBE" while the maximum confusion rate of the proposed method is 8.0% for "DIREITA" misrecognizing as "ESQUERDA".

These differences point to a better reliability of the proposed method. Also, it is important to note that the proposed method uses lower computational effort than the BMC method – the proposed method deals with 158 input units, while the BMC method deals with 1280 input units.

3.3. Ongoing work

Authors believe that better results can be obtained with the increase of independent speaker voices database, and with a better refinement of the used preprocessing algorithms. Ongoing works previews the use of a larger voice command database, noise robustness studies, and test other wavelet standards.

References

[1] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques", *IEEE Trans. Acoustics, Speech and Signal Proc*., v. Assp-27, pp. 336-349, Aug. 1979.

[2] F. Itakura. "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Trans. Acoustics, Speech and Signal Proc*., v. ASSP-23, pp. 57-72, Feb. 1975.

[3] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi. "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", *Journal of Computing*, v. 2, n. 3, pp. 138-143, Mar. 2010. ISSN 2151-9617.

[4] R. Gandhiraj, P.S. Sathidevi. "Auditory-Based Wavelet Packet Filterbank for Speech Recognition Using Neural Network", *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, 2007, pp. 666-673.

[5] Kidae Kim, Dae Hee Youn, and Chulhee Lee. "Evaluation of wavelet filters for speech recognition", *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, 2000.

[6] José Antônio Martins. *Avaliação de Diferentes Técnicas para Reconhecimento de Fala*. PhD Thesis, Universidade Estadual de Campinas, 1997.

[7] Robert Modic, Børge Lindberg, and Bojan Petek. "Comparative Wavelet and MFCC Speech Recognition Experiments on the Slovenian and English SpeechDat2", *ISCA tutorial and research workshop on non-linear speech processing*, Le Croisic: France, May 2003.

[8] I. Daubechies. *Ten lectures on wavelets, CBMS-NSF conference series in applied mathematics* (SIAM Ed., 1992), pp. 6.

[9] Audacity software, version 1.2.6, downloaded in May-19-2009 at *http://audacity.sourceforge.net/*

[10] I. S. Peretta, G. F. M. Lima, J. A. Tavares, K. Yamanaka. "Reconhecimento de Comando de Voz Baseado em Filtros Wavelet Utilizando Redes Neurais Artificiais. In: *IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional*, Ouro Preto: Brazil, 2009.

[11] J. F. Kaiser. "Some useful properties of Teager's energy operators", *IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP-93), vol.3*, 1993, pp. 149-152.

[12] A. V. Oppenheim, R.W. Schafer, and J.R. Buck. *Discrete-time signal processing* (Prentice-Hall Inc., 2nd edition, 1998), pp. 465-473.

[13] L. Fausett. *Fundamentals of neural networks* (Prentice-Hall Inc., 1994), pp. 289-304.

[14] L. R. Rabiner, and M. R. Sambur. "An Algorithm for Determining the Endpoints for Isolated Utterances", *The Bell System Technical Journal*, v. 54, n. 2, Feb. 1975, pp. 297-315.

[15] L. F. Lamel, L. R. Rabiner, A. E. Rosemberg, and J. G. Wilpon. "An Improved Endpoint Detector for Isolated Word Recognition". *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29(4):777-785, Aug 1981.

[16] Martin F. Møller. "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", *Neural Networks*, v.6, n. 4, pp. 525-533, 1993.

[17] Gregory Levitin. *Computational Intelligence in Reliability Engineering*, in: Studies in Computational Intelligence, v.39, pp. 385-386, Ed. Springer Berlin Heidelberg New York, 2007. ISBN 978-3-540-37367.