An Overview of Basics Speech Recognition and Autonomous Approach for Smart Home IOT Low Power Devices

Jean-Yves Fourniols, Nadim Nasreddine, Christophe Escriba, Pascal Acco, Julien Roux, Georges Soto-Romero

Journal of Signal and Information Processing, 2018, 9, *-*
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

An Overview of Basics Speech Recognition and Autonomous Approach for Smart Home IOT Low Power Devices

Jean-Yves Fourniols, Nadim Nasreddine, Christophe Escriba, Pascal Acco, Julien Roux, Georges Soto Romero

Laboratory for Analysis and Architecture of Systems, LAAS, University of Toulouse, Toulouse, France
Email: fourniol@laas.fr

How to cite this paper: Author 1, Author 2 and Author 3 (2018) Paper Title. *Journal of Signal and Information Processing*, **9**, *-*.
https://doi.org/10.4236/***.2018.*****

Received: **** **, ***
Accepted: **** **, ***
Published: **** **, ***

Abstract

Automatic speech recognition, often incorrectly called voice recognition, is a computer based software technique that analyzes audio signals captured by a microphone and translates them into machine interpreted text. Speech processing is based on techniques that need local CPU or cloud computing with an Internet link. An activation word starts the uplink; "OK google", "Alexa", … and voice analysis is not usually suitable for autonomous limited CPU system (16 bits microcontroller) with low energy. To achieve this realization, this paper presents specific techniques and details an efficiency voice command method compatible with an embedded IOT low-power device.

Keywords

Voice Recognition, Speech Processing, Voice Command, Embedded Device

1. Introduction

Human Machine interface based on speech recognition systems is a reality made possible through an Internet link and multi-threaded, multi-pipelined processor architecture or open source applications. This paper aims to analyze the development of a low cost and low power speech recognition system. The main challenge in this project is to realize the speech recognition system on embedded hardware, using limited resources (computing power, embedded energy) and based on a very small microcontroller (16 bits). This is a difficult task taking into account that a speech recognition system requires high processing power for the audio signal treatment [1] [2].

The developed system is able to successfully distinguish and recognize short

basic voice commands composed from a few words. Also, the language is used for recognition it doesn't matter, the accuracy and reliability of the system remain almost the same in every case. The system is designed to be speaker independent, so it is capable in recognizing voice commands spoken by different persons.

The goal of the system is to help people with disabilities in making their lives easier, by letting them control different things only with voice commands. As well, it can be implemented to simplify the usage of different appliances which have too many hardware buttons for a high number of inputs [3] [4].

This paper is divided in five main parts. These describe the state of art, then the analytical description of the system, followed by the algorithm description, and the recognition technique to conclude with results of the recognition.

2. Overview State of the Art

Speech recognition appeared in 1950 when the first digit recognition system was developed, a fully wired device and very unreliable. By 1960, the introduction of numerical methods and computer usage had entirely change research dimension.

However, the results were very poor because everyone had largely underestimated the realization difficulty of the whole system, particularly for the continuous speech type of recognition system [5] [6].

Around 1970, the need to use linguistic constraints in automatic speech decoding had been regarded as an engineering problem [7]. But in the end of the 70s the first generation of speech recognition system based on isolated words started to be commercialized.

The following generations have started to take advantage of the increasing and increasing power calculation of the computers [8], showing very promising results [9]. "Dragon speaking" is one of the best computer software in speech recognition commercialized today. Nowadays, "OK Google", "Alexa", "Siri" and "S-voice" services offered by Apple and Samsung prove to have very good speech recognition accuracy on their mobile devices [10].

Most publications show the usage of this recognition is computer-based systems [6] [9] [10]. Many embedded software exist but they need a high computing power like 32/64 bits microcontrollers or a Raspberry Pi [11]. Few of them propose this integration on a limited embedded system but, actually, the voice recognition is deported [12] or the system has a high consumption [13]. Our following algorithm is designed to be implemented in a power-limited system by limiting the calculation time and the energetic consumption.

In general, speech recognition systems are devised in 3 important stages as follows:

- Audio capture: a transducer (e.g.: microphone) that captures the audio signal, when a user is talking, and transforms it in electrical signal
- Sound analysis and parameterization: it will analyze, decode and parameterize the audio signal captured by the sensor. This step is a mathematical

treatment of signal and it is done in time, frequency and intensity domains. Here the audio signatures of the words will be extracted from the actual audio signal.

- The identification: the decision in choosing the right spoken voice commands is done in this step. Basically, here the program will compare the audio signatures of the speech commands spoken by the user with the ones already learned (stored) in the system.

Figure 1 resumes these three stages on a schematic.

2.1. Voice Characteristics

Human voice properties should be taken in account in developing a speech embedded recognition system:

- The bandwidth of the speech signal is around 4 kHz.
- The speech signal is periodic and has a fundamental frequency between 80 Hz and 350 Hz.
- Peaks exist in the spectral distribution of energy of the voice signal. The frequencies around these values are called formant frequencies.

$$F_{\text{peaks}} = (2n-1)\times 500\text{Hz with } n = 1,2,3,\cdots \quad (1)$$

- Depending on the shape of the vocal tract the frequency of the formants, especially the first and second, will change, therefore they will characterize the way vowel are articulated.
- The envelope of the voice power spectrum also decreases with the increase of frequency with about 6 dB per octave.

2.2. Parametrization

First step is to configure the speaker's voice signal looking for a "signature" to be founded for recognition. In order to do this, several methods exist.

First type consists of spectral analysis. It is based on the frequency decomposition of the signal without a prior knowledge of its fine structure. The best and most used method is the one using Fast Fourier Transform (FFT), more precise the Discrete Fourier Transform calculation (DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{i2\pi nk}{N}} \quad (2)$$

Applying the DFT to a complex sound, and repeating this procedure, a graphic will be drawn showing time amplitude and frequency evolution, as we can see at **Figure 2**. This will define the sound audio signature. Specific characteristics are extracted and used from this calculation, or even the whole result, in the form of vectors or matrix later in the processing and identification stages.

The second method consists on identification by understanding the mechanisms of sound production. The most commonly used approachis based on linear predictive coding (LPC). The basic idea is that the mouth channel is constituted by a cylindrical tube with varying sections. The adjustment of the parameters of this model allows determining at almost any moment the transfer function. Af-

terwards, this provides an approximation of the spectrum envelope of the audio signal at the instant analysis. Then, it easily identifies the formant frequencies, with the help of the resonant frequencies of the vocal tract. They correspond to the maximum energy in the spectrum. By repeating this method continuously, the audio signature of the sound will start to show. A LPC representation is shown in **Figure 3**.

Once the audio signature is obtained, the speech recognition procedure can move to the next step.

2.3. Isolated Word Recognition

Speech recognition systems can be configured to work on isolated words or even on continuous speech [14] [15]. The most used is the one on isolated words because it has the highest rate of accuracy and also it doesn't require a powerful hardware as the complex method of continuous speech does, making it suitable for a low budget system. The absence of indicators in speech signal for the boundaries of phonemes and words is a major difficulty in speech recognition. Thus pronouncing words with an artificially isolation, a small silence exceeding a few tenths of a second, in speech commands represents a significant simplification of the problem.

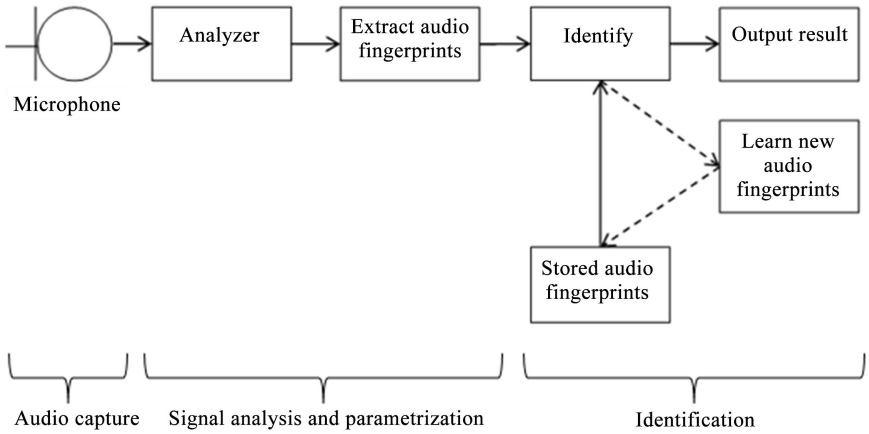

Figure 1. Stages in a speech recognition system.

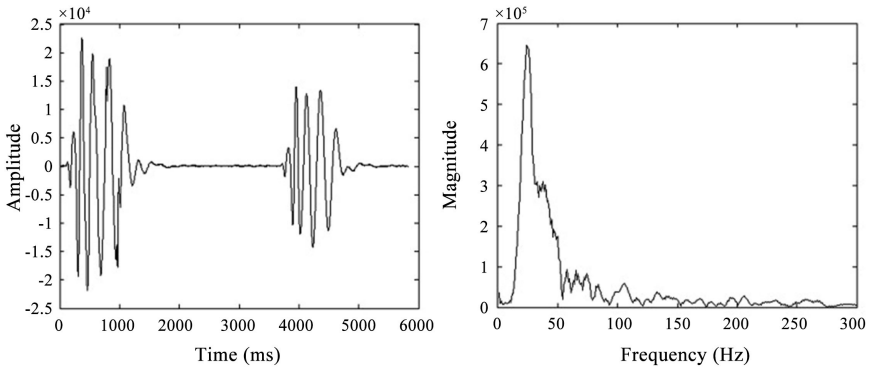

Figure 2. DFT calculation of an audio signal.

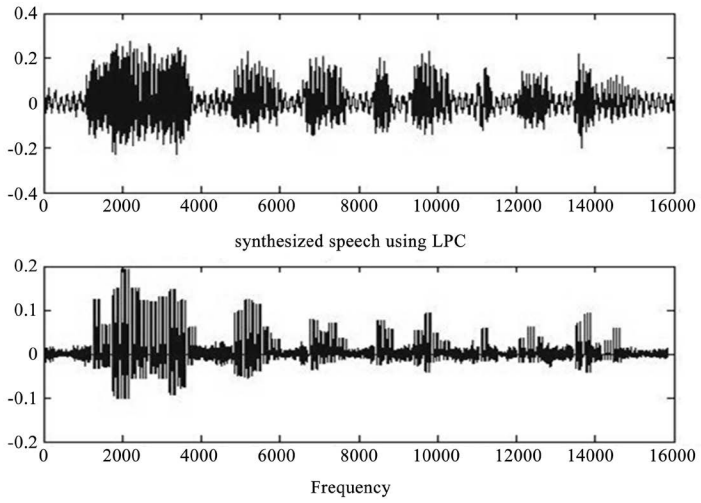

Figure 3. LPC calculation of an audio signal.

Two types of this system currently exist:

- The speaker dependent system—it can be used only by one user and it needs to be trained. A person should dictate a set of words, which maximizes the recognition rate and extend the vocabulary used. The disadvantage is that it can be used by one person only.
- The speaker independent system—it uses a database containing averages of audio signatures allowing the recognition of speech commands spoken by different persons. The main drawback is that the system is not equipped with learning capabilities and the number of words limited.

To increase the recognition rate of the system, by making it work even if the person speaks on different tonalities (different octaves) or if more than one person is used, a normalization process must be implemented. This will be done before the system will start to decode voice commands into phonemes, syllables or words depending on the technique of the system.

2.4. Recognition Techniques

Recognition technique is based on two approaches, the global and the analytical method [16] [17].

In the global approach (entire word), the basic unit is often the word seen as a global entity that is not decomposed. The idea of this method is to give the system an acoustic image of each word that will have to identify it later. This operation is done during the training phase, were each word is pronounced one or more times. This method has the advantage of avoiding the effects of articulation. It is however limited to small vocabulary by a limited number of speakers.

The analytical approach (structure of the word), which takes advantage of the linguistic structure of words, attempts to detect and identify the basic components (phonemes and syllables). These are the basic units to recognize. This method is simpler because only the features of the base units, instead of the whole words, have to be registered in the memory.

In fact, both approaches basically are the same; the difference is the entity to be recognized, "the word" for the first and the "phoneme" or syllable for the second.

2.5. Working Principle

The structure of the isolated word speech recognition system can be distinguished in two phases:

The training phase—a user dictates the entire vocabulary used in the voice commands in order to create the reference audio signatures of the commands. But for the analytical the user will only dictate some specific words which contain important successions of phonemes. For an independent speaker system, this does not exist.

The recognition phase—the user says the actual voice command which contain the words from the stored vocabulary. Then, the word recognition system is typical problem of pattern recognition.

The calculation done in the recognition phase, when comparing speech commands, is not that simple because words can have different forms depending on the user and speech rate. A speaker cannot pronounce several times the same speech sequence with exactly the same rate and same duration.

Also, time alignment is a problem because the user cannot repeat the same speech commands with the same pause between the words. It is very important that a special time warping algorithm is implemented in order to manage this problem.

Comparison methods by dynamic programming have been widely used for recognition of isolated words. The most commonly calculations used for this method are: the Euclidean squared distance, hidden Markov models and neuro-mimetic models. Classifying the calculation techniques after the required processing power, the Euclidean squared distance needs the least of them all. This makes it suitable for every speech recognition system which has a limited hardware budget. The Euclidean squared distance it is the simplest way to determine the similarity between words in speech commands. If the parameterization is done correctly, then the results obtained with this formula can have very high rate accuracy in the identification of the right spoken voice command. To be more precise, with this formula it can be calculated the actual difference between two vectors containing different audio DFT results for example [18].

3. Our Methodology

With the main characteristics of speech recognition systems presented in the previous paragraphs, the challenge is to put all of those features into a low cost hardware and energy consuming. The system is configured to work on recognizing isolated words based on the global technique, the words being the unit for recognition. The words in the voice commands must have a small pause between them. As well, they are also aligned and delimited by the system. Audio signa-

tures are extracted with the DFT spectral method.

With the help of the normalization process implemented, the system can be considered speaker independent, even though, before usage, the system has to be trained with the actual voice commands that will be used afterwards. Also, by using normalization process, the successful recognition rate is increased.

Users are able to use and store on our system a defined number of voice commands. A voice command has a defined length of 2 seconds, enough for a person to say a few keywords for a command. For the speech recognition system, it's important to be able to distinct commands even though they are said on different tonalities and by different users, thus a normalization process is implemented in the system. This is done on every signature that is stored and that is currently being processed for command identification. Sometimes the speech commands have the same words in their composition. To avoid confusion between commands the system does the identification routine for the whole sentences and individually for each corresponding word in the sentences. Then it compares on how many words have been identified from the spoken command with the ones in a stored voice command. So, the voice command with the most identified words it's taken in account. The Euclidean squared distance formula is used to calculate the similarities between the audio signature of a spoken command and the audio signatures of the stored voice commands. Depending on which result is the smallest or which one is under a predefined level, the right voice command is recognized.

$$d(c,s) = d(s,c) = \sum_{i=1}^{n} (c_i - s_i)^2 \tag{3}$$

d—distance (*similarity between commands*)

c—current spoken voice command

s—stored voice command

n—number of sound signatures (*time slots*) *in a voice command*

The 8-kHz frequency sampling rate it's chosen because it will offer a frequency range between 0 and 4 kHz and it will match the human voice, which has a frequency range from 300 Hz to 3.3 kHz. A sampling frequency beyond that value will be useless. **Figure 4** presents a vocal command represent relative to the time and **Figure 5** presents four different DFT applied on this signal.

As it can be seen from the DFT simulation (**Figure 5**) in the majority of the cases, every time 2 or 3 important frequency peaks with big density stand out. Also, this thing can vary by a little bit, depending on the language that is spoken. In English 3 peaks stand out.

Because the DSP engine in the microcontroller is optimized to do a 256-point length type DFT [19] [20], this spectral solution is chosen to extract the voice commands signatures. For an incoming audio signal, with a sample rate of 8 kHz, the 256 point length DFT is done every 32 ms, thus for a time window of 2 seconds are obtained about 64 DFT results, from which the voice commands signatures can be extracted.

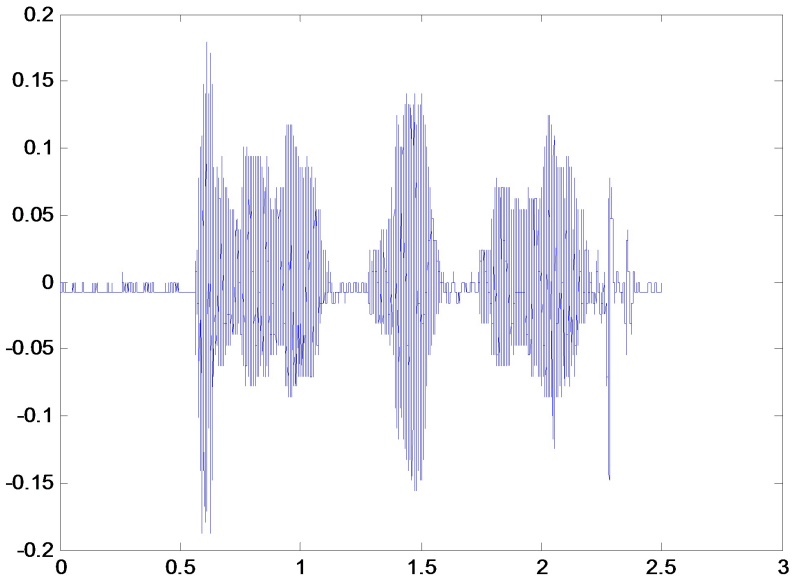

Figure 4. Voice command "Open the window" composed of 3 words.

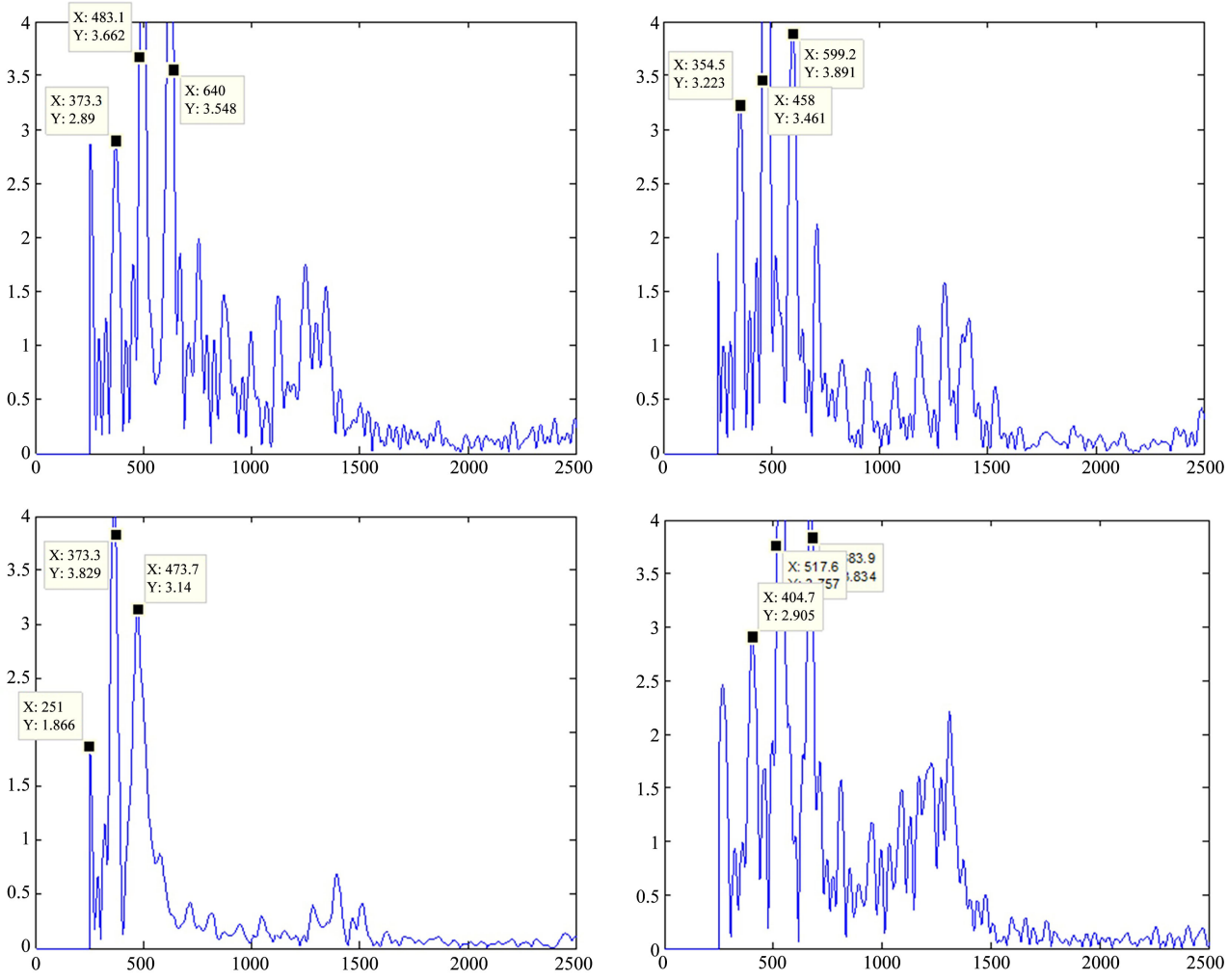

Figure 5. Different 32 ms DFT time windows of a voice command.

$$\mathrm{DFT}_{\mathrm{length}} \div \mathrm{Frequency}_{\mathrm{sample}} = \mathrm{DFT}_{\mathrm{period}}$$

$$256\mathrm{points} \div 8000\mathrm{Hz} = 32\mathrm{ms}$$

$$2048\mathrm{ms} \div 32\mathrm{ms} = 64\mathrm{DFT\ results} \quad (4)$$

The sound signatures with its three important high peaks define the audio signature. These are gathered from each successive 32 ms DFT calculation for a time of 2 seconds resulting in the final voice command audio signature.

To compare, the execution time of a DFT in a not-DSP microcontroller is very long. This time depends of the length of the processed data. For 8 Mhz clocked microprocessor, the execution time of an implementation of a DFT can be calculated by a polynomial expression.

$$t_{\mathrm{execution(ms)}} = 0.065N^2 + 0.2937N + 2.8 \quad (5)$$

For a data of 256 points, the execution lasts 4338 ms which consumes 28.3 μA.h. A DSP, for the 256 points DFT, consumes 0.18 μA.h. For our application, only the DSP can be chosen because of the slowness of the regular microcontroller. Moreover, the electric consumption is reduced by 150, another advantage of using the DSP.

For our algorithm on a DSP, a recognition of a three-second-long text costs 17.28 μA.h. This consumption is compatible with embedded systems. For example, using a button cell CR-2032 (210 mA.h), we can recognize 972 sentences of 3 seconds that gives an equivalent autonomy of more than a month in continuous mode.

4. Algorithm Description

The analog to digital converter is set to process the data captured by the microphone at a sampling rate of 8 kHz and to transfer it into a temporary buffer. Every time this buffer is filled, it triggers a function which starts to do the 256-point length DFT calculation.

At this DFT length and audio sample rate, one unit (frequency bin) of the resulted DFT calculation has a range of frequency of 31.25 Hz. This bin represents the frequency resolution and it contains the actual magnitude of the audio signal in that specific frequency range.

$$\mathrm{Frequency}_{\mathrm{sample}} \div \mathrm{DFT}_{\mathrm{length}} = \mathrm{Frequency}_{\mathrm{resolution}}$$

$$8000\mathrm{Hz} \div 256\mathrm{points} = 31.25\frac{\mathrm{Hz}}{bin} \quad (6)$$

The whole result of the calculation is stored in a vector with a length of 128 values. This contains all the frequency bins and has a frequency range of 0 - 4 kHz. Because this range is too wide for the human voice, it is cropped into a 91-length vector, which corresponds to the range of 280 Hz to 3 kHz. Also, this crop is done in order to eliminate the noise from the low frequency spectrum, caused sometimes by the microphone. The obtained vector represents the magnitude of the raw audio signal in that frequency range for a time of 32 ms.

All DFT calculations and most vector manipulations are done with the DSP engine, integrated in the microcontroller. This vastly reduces the processing time and frees the CPU workload.

In the following step, 3 frequency bins, which have the highest magnitude and represent the highest peaks of the audio signal, are extracted from the resulted vector of the DFT calculation. It is important that, between the selected frequency bins, a distance of at least 3 bins (93.75 Hz) exists, so the system will not pick up values from the same frequency peak.

After the 32 ms sound signature is created, the system is verifying in a loop, for every DFT calculation, if the magnitude of the highest frequency bin is different from 0. In this way, the system knows if something has been spoken by the user and that it is ready to proceed to the following processing steps.

If the system has detected any sign of voice, it starts to record, in a "First-in-First-Out" algorithm, the 3 important high peaks of the 32 ms DFT calculation, for a time length of 2 seconds. For 2 a second length and a 32 ms DFT, the FIFO process will store in total 192 values, which means 64 values for each 3 important frequency peaks. In the same, the LEDs on development board will turn on for 2 seconds to show the user that the audio data is being recorded and also to help him fit his voice command in that time window.

After the recording stops, the values from the FIFO algorithm are distributed in order into 3 separate vectors, each one having a length of 64 values. In one vector are stored the values containing the first highest peak frequency bins, in the second vector are stored the other values containing the second highest peak frequency bins, and in the final vector are stored the third highest frequency bins. These 3 vectors represent the audio signature of the spoken voice command.

Because everyone differs in how they speak, by pronouncing the words at different frequencies and magnitudes, and in order to make the system a speaker independent one, the 3 vectors are normalized. The normalization process is done separately for every vector. It is done by calculating the average of all values from a vector and then by dividing each value with the calculated average.

In the next step the vectors are passed to a time warping process. This process fixes the words length to 12 values (time slots), in each vector, and separates the words at a defined distance. The system is configured to detect and identify a word in a vector, if more than 5 consecutive values of 0 exist after 2 consecutive values different from 0. In this way, the vectors containing the audio signature of the voice command have their words synchronized and can be compared with others.

As the audio signature of the 2-second-long voice command is now processed, the system moves to the next step of comparing the current audio signature with the ones already stored on the flash memory.

If the system has not been trained, no audio signatures are stored on the flash memory; it will just compare the current audio signature with blank audio sig-

natures, resulting in an unidentified voice command. If it has commands stored, it will compare the current signature with stored ones.

The comparison between the audio signatures is done by different techniques using the Euclidean squared distance. Depending on the results, the current voice command is identified, or not, with one of the stored voice commands.

The speech recognition system is configured to have a master voice command in order to prevent the system from mistakenly recognizing different voice commands. This command activates the system for a recognition session, a time window of 10 seconds, in which the user can say his actual voice commands. After the 10 second timer expires, the system deactivates and the user is obliged to repeat the master command in order to resume. The master voice command has the same properties, as the rest of the commands, and it also needs to be stored in the training phase, just like the others.

Finally, after all the processing and calculations are done, the user can now choose to store his desired voice command in order to train the system, if no audio signatures are stored on the flash memory, or to retrain the system, if the user is not satisfied with the already stored voice commands.

5. Recognition Technique

The technique used in recognizing the voice commands is based on the Euclidean squared distance. This formula can be applied in many ways between the audio signatures of the voice commands, but after many experiments, the following two calculation methods have been chosen and they are presented in the **Figure 6**:

Global distance—this is done by calculating the distance between the whole vectors of the audio signatures. After the 3 vectors are processed, containing first, second and third highest peaks of the 64 time slots of a voice command, they are ready for the distance calculation. First calculation is done between the first vector, containing the first highest peak, of the current processed voice command and the corresponding first vector of a stored voice command. This continues by calculating in the same way for the second and third vectors. In the end three values are obtained representing the distances between the first, second and third vectors of two voice commands. These values are then averaged in ordered to obtain the final distance/difference between vectors.

This calculation is done individually between the current voice command and each voice command stored. If one of the final results is under the value 150 and it's the smallest from the rest of the final values, then the voice command stored, corresponding to the obtained result, is considered the recognized command.

Word distance—this is done by calculating the distance between the words from vectors of the audio signatures. This method was chosen, in addition to the previous one, in order to avoid confusions made by the system in the scenario when the voice commands contain the same words. Also, as the previous method, this calculation is done after the vectors containing the audio signatures

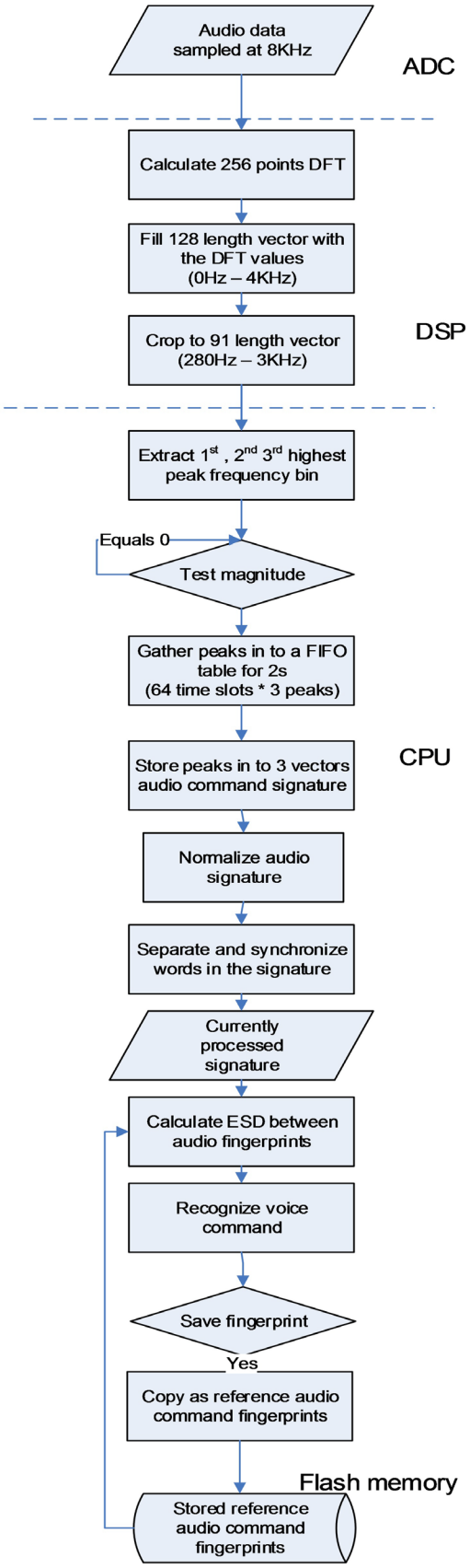

Figure 6. Algorithm description of the speech recognition system.

are processed. This method begins by calculating the distance between the first word from the first vector, of the current processed voice command, and the corresponding first word from the first vector, of a stored voice command. Then, it continues by calculating in the same way for the next words in the first vector of the voice command, obtaining in total five word distances.

The calculation is done for the next two vectors, resulting in another ten values and in total fifteen values. These are then averaged, taking in account the words they correspond to, obtaining 5 final distances.

As the previous method, this calculation is done individually between the current voice command and each voice command stored. Depending on which voice command has the biggest number of smallest final distances between words that is considered the recognized voice command.

After calculating both methods in parallel, the voice commands chosen and recognized by the calculation methods are compared in order to make the right decision, resulting in three cases as follows:

- Case 1: If both voice commands identified by the two methods match, then the resulted voice command is recognized.
- Case 2: If the voice commands identified by two methods do not match, then the system will partially recognize the voice command resulted from the global Euclidean squared distance method and it will ask the user to repeat the voice command.
- Case 3: If the value resulted from the global Euclidean squared distance method it's above 150, it will not identify anything and the result from the second calculation method will not be taken in account anymore, forcing the system to not recognize any voice command at all.

Table 1 resumes the results of these three cases.

6. Results

The developed speech recognition system was tested in order to calculate its accuracy and reliability. Tests were done with English language, a worldwide language [21] [22]. We present here English language, in a quiet and in a noisy environment, with one person and with two persons. The system was trained with the following three voice commands spoken ten times:

Table 1. Voice commands recognition technique.

	Method I (global) result - priority	Method II (word) result	Final result
Case 1	voice command "a" recognized	voice command "a" recognized	voice command "a" recognized => recognition
Case 2	voice command "b" recognized	voice command "c" recognized	voice command "b" partially recognized => confusion
Case 3	No voice command recognized (>150)	Not taken in account anymore	No recognition

- 1st voice command: "Turn on the light"
- 2nd voice command: "Close the window"
- 3rd voice command: "Open the door"

6.1. Speech Test in a Quiet Environment

Output results from the two recognition methods for the first voice command "Turn on the light" are presented in **Figure 7** and in **Figure 8**.

It can be observed **Figure 7** that the spoken voice command has the smallest value every time and it's easy to take decision in recognizing the right voice command. A small exception being in the 5th case, when the distance is above the minimum required value of 150 and the system will not recognize anymore the voice command.

For the same spoken voice command, but now with the second method, it can be observed that the spoken voice command has the biggest amount of recognized words every time. So, taking in account that the first method has priority, and by combining the results, it turns out that accuracy of the system for the first spoken voice command is 90%.

Figure 9 and **Figure 10** presents the results with the second voice command "Close the window".

It can be noticed that not each the time the spoken voice command has the smallest value in this chart, so in order to improve the accuracy it has to be taken in account the second method. Also, in 2nd and 6th case the values are above 150, so they are not taken in account anymore. Worse, in the 5th case another voice command has the smallest value, decreasing even more the recognition rate.

The second method for the same spoken voice command helps in clarifying which test number has the biggest number of recognized words and confirms the results from the first method. The final successful recognition rate is 70% for this voice command.

Output results from the two recognition methods for the third voice command "Open the door" are shown in **Figure 11** and **Figure 12**.

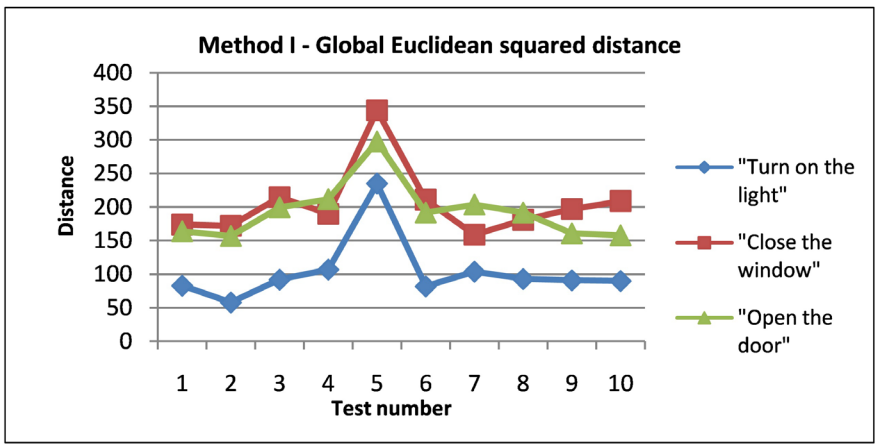

Figure 7. "First training command" first result.

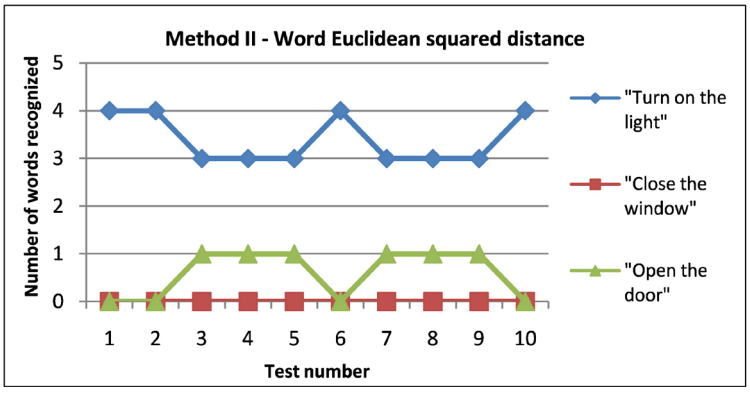

Figure 8. "First training command" second results.

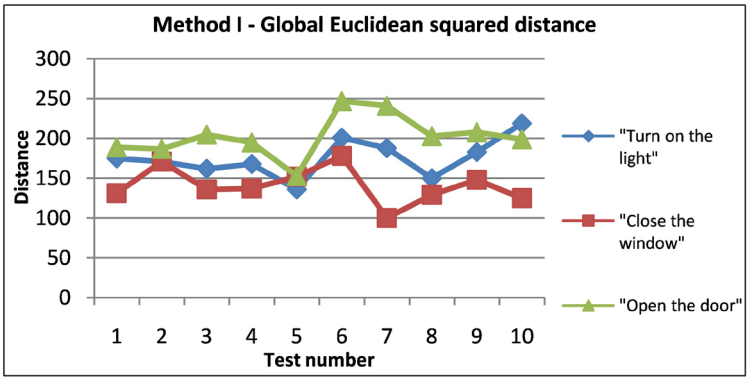

Figure 9. "Second training command" first results.

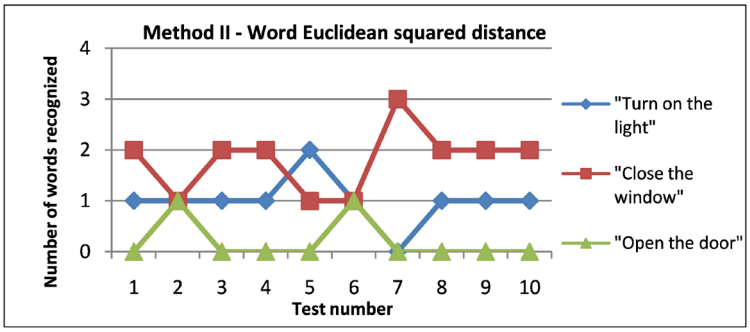

Figure 10. "Second training command" second results.

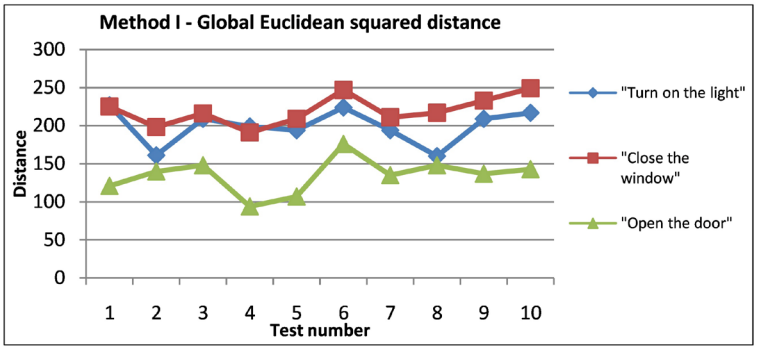

Figure 11. Third training command first results.

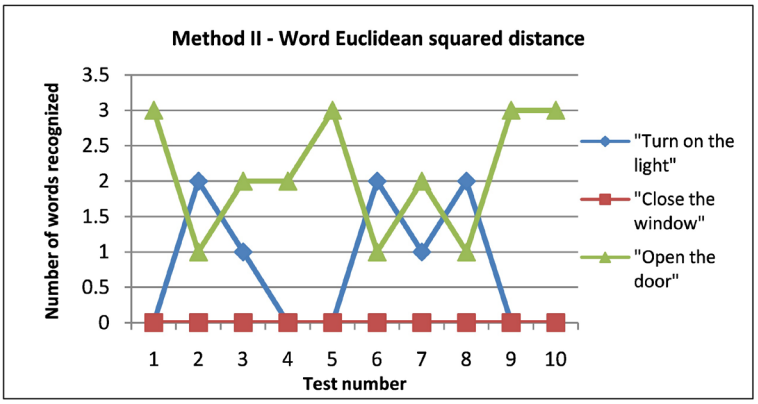

Figure 12. Third training command second results.

In this chart, it can be seen that the spoken voice command has all the time the smallest value. Even though, the distance is pretty big in the 6th test, being above 150, it's easy for the system to take a decision in recognizing the right command. Now it only needs confirmation from the second method.

The second method confuses three times the spoken voice command with another command but it confirms for the rest of the tests. This is the reason why the first method has priority in the final decision. The recognition rate for this command is 70%.

The final results depend very much on how the user pronounces the actual voice command in comparison with the stored voice command. This is the reasons why is better to have two recognition methods and to compare their results in order to take the final decision in recognizing the voice commands

After reviewing and combining all the results in **Table 2**, obtained from the tests done, it can be concluded that the speech recognition system has achieved successful recognition rate of 90.0%.

6.2. Speech Test in a Noisy Environment

To test how the system will perform in a noisy environment, the system was trained in a quiet environment with the same English voice commands from the previous test. Then, the voice commands were spoken ten times each by the same person, but in a noisy environment this time.

The environment noise consisted from white noise and a couple of music files. They were played through a pair of loudspeakers which are capable of outputting 4 watts of power. **Table 3** presents the results in this noisy environment.

After the test was completed and all the results were analysed, the system showed a 85% accuracy in recognition, a little bit under accuracy showed in the test with a quiet environment.

6.3. Speech Test with Different Persons

To test how it will perform as speaker independent system, the system was

Table 2. Test gathered results.

Voice commands	Good recognition	Bad recognition	Confusion	No recognition
"Turn on the light"	90%	5%	5%	0%
"Close the window"	90%	5%	0%	5%
"Open the door"	90%	5%	5%	0%
Total successful recognition: 90.0%				

Table 3. Test gathered results in a noisy environment.

Voice commands	Good recognition	Bad recognition	Confusion	No recognition
"Turn on the light"	85%	5%	10%	0%
"Close the window"	85%	5%	10%	0%
"Open the door"	85%	5%	10%	0%
Total successful recognition: 85%				

Table 4. Test gathered results with different persons.

Voice commands	Good recognition	Bad recognition	Confusion	No recognition
"Turn on the light"	85%	10%	5%	0%
"Close the window"	85%	5%	10%	0%
"Open the door"	85%	5%	10%	0%
Total successful recognition: 85%				

trained again in a quiet environment with the same English voice commands from the first test. Then, in the recognition phase, the voice commands were repeated ten times each in a quiet environment, but by a different person this time. **Table 4** shows the recognition rate for this test.

Analyzing the obtained results, the speech recognition system showed still maintaining a good reliability, even though different persons were used for the training phase and recognition phase.

7. Conclusion

The developed speech recognition system has performed with an almost identical accuracy with few words for several users. This system has equivalent results in a quiet and in a noisy environment. It can support different persons too. So this system can be easily deployed in a house. Further, the system can be adapted to another language by changing its processing parameters, like the number of time slots reserved for every word.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kamarudin, M.R., et al. (2013) Low Cost Smart Home Automation via Microsoft Speech Recognition. *International Journal of Engineering & Computer Science IJECS*, *IJECS-IJENS*, **13**.

[2] Kumar, P., Deshmukh, M. and Kumar, A. (2018) A Novel Pitch Based Voice Recognition Model (PVRM). 1-4.

[3] Georgoulas, C., Raza, A., Güttler, J., Linner, T. and Bock, T. (2014) [illegible]C. *Proceedings of the International Symposium on Automation and Robotics in Construction*; Vilnius Vol. 31, 1-9. Department of Construction Economics & Property, Vilnius Gediminas Technical University, Vilnius.

[4] [illegible] (1993) Fundamentals of Speech Recognition. [illegible]://volyubemw.updog.co/

[5] Becchetti, C. and Ricotti, K.P. (2008) Speech Recognition: Theory and C++ Implementation (With CD). John Wiley & Sons.

[6] Thiang, D.W. (2009) Limited Speech Recognition for Controlling Movement of Mobile Robot Implemented on ATmega162 Microcontroller. *International Conference on Computer and Automation Engineering.*

[7] Amodei, D., Ananthanarayanan, S. and Anubhai, R. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *Proceedings of Machine Learning Research*, **48**.

[8] [illegible]ndigarh Engineering College, Landran (Mohali), India and et al. Verma (2018) Automatic Speech Recognition Using Mel-Frequency Cepstrum Coefficient (MFCC) and Vector Quantization (VQ) Techniques for Continuous Speech. *International Journal of Advanced and Applied Sciences*, **5**, 73-78.

[9] Kumar, S. (2014) Ubiquitous Smart Home System Using Android Application. *International Journal of Computer Networks & Communications*, **6**, 33-43.

[10] Sonia, B. and Sridhar, D.S. (2018) Implementation of Voice Recognition Technology in Hospitals Using Dragon NaturallySpeaking Software. *Biometrics and Bioinformatics*, **10**.

[11] Vojtas, P., Stepan, J., Sec, D., Cimler, R. and Krejcar, O. (2018) Voice Recognition Software on Embedded Devices. In: Nguyen, N.T., Hoang, D.H., Hong, T.-P., Pham, H. and Trawiński, B., Eds., *Intelligent Information and Database Systems*, Vol. 10751, Springer International Publishing, Cham, 642-650.

[12] Lai, C.-H. and Hwang, Y.-S. (2018) The Voice Controlled Internet of Things System. 1-3.

[13] Basyal, L., Kaushal, S. and Singh, G. (2018) Voice Recognition Robot with Real Time Surveillance and Automation. *International Journal of Creative Research Thoughts* (*IJCRT*), **6**, 11-16.

[14] Newman, M.J. *et al.* (2015) Embedded System for Construction of Small Footprint Speech Recognition with User-Definable Constraints. Patent US9117449B2,

2015-08-25.

[15] Perez-Cortes, J.C. and Guardiola, J.L. (2009) Pattern Recognition with Embedded Systems Technology: A Survey. 20*th International Workshop on Database and Expert Systems Application.*

[16] Jiang, H., *et al.* (2014) Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio*, *Speech*, *and Language Processing*, **22**, 1533-1545.

[17] Deng, L. and Li, X. (2013) Machine Learning Paradigms for Speech Recognition: An Overview. *IEEE Transactions on Audio*, *Speech*, *and Language Processing*, **21**, 1060-1089. https://doi.org/10.1109/TASL.2013.2244083

[18] IEEE (2012) IEEE Systems, Man, and Cybernetics Society. *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part C* (*Applications and Reviews*), **42**, C2.

[19] Sinha, P. (2009) CPU Architectures for Speech Processing. In: *Speech Processing in Embedded Systems*, Springer International Publishing, Boston, MA, 55-74

[20] Nedevschi, S., Patra, R.K. and Brewer, E.A. (2005) Hardware Speech Recognition for User Interfaces in Low Cost, Low Power Devices. *Proceedings of* 42*nd Design Automation Conference*, Anaheim, CA, USA, 13-17 June 2005.

[21] Schafer, E.C., *et al.* (2017) Speech Recognition in Noise in Adults and Children Who Speak English or Chinese as Their First Language. *Journal of the American Academy of Audiology*. https://doi.org/10.3766/jaaa.17066

[22] Li, K., Mao, S., Li, X., Wu, Z. and Meng, H. (2018) Automatic Lexical Stress and Pitch Accent Detection for L2 English Speech Using Multi-Distribution Deep Neural Networks. *Speech Communication*, **96**, 28-36. https://doi.org/10.1016/j.specom.2017.11.003