

DSP Real-Time Implementation of an Audio Compression Algorithm by using the Fast Hartley Transform

Souha BOUSSELMI, Noureddine ALOUI and Adnen CHERIF
University of Tunis El-Manar, Science Faculty of Tunis
Laboratory of Analysis and Processing of Signals
And Electric and Energy System, 1060 Tunis, Tunisia

Abstract—This paper presents a simulation and hardware implementation of a new audio compression scheme based on the fast Hartley transform in combination with a new modified run length encoding. The proposed algorithm consists of analyzing signals with fast Hartley Transform and then thresholding the obtained coefficients below a given threshold which are then encoded using a new approach of run length encoding. The thresholded coefficients are, finally, quantized and coded into binary stream. The experimental results show the ability of the fast Hartley transform to compress audio signals. Indeed, it concentrates the signal energy in a few coefficients and demonstrates the ability of the new approach of run length encoding to increase the compression factor. The results of the current work are compared with wavelet based compression by using objective assessments namely CR, SNR, PSNR and NRMSE. This study shows that the fast Hartley transform is more appropriate than wavelets one since it offers a higher compression ratio and a better speech quality. In addition, we have tested the audio compression system on DSP processor TMS320C6416. This test shows that our system fits with the real-time requirements and ensures a low complexity. The perceptual quality is evaluated with the Mean Opinion Score (MOS).

Keywords—Speech compression; Fast Hartley transform (FHT); Discrete Wavelet Transform (DWT)

I. INTRODUCTION

The advancement of communication technology and the growth of the Internet have made speech compression a prime concern in the field of digital signal processing. The main motivation behind development of speech compression schemes is to reduce the number of bits required to represent an audio signal in order to minimize memory storage costs and transmission bandwidth requirements. The fundamental way of audio compression is based on removing signal redundancy while preserving the intelligibility of the signal. An audio compressor is characterized by three factors which are reconstructed audio quality, the amount of data compression and the codec complexity. There is always a compromise between increasing the compression ratio, maintaining the quality and intelligibility of the reconstituted voice. Compression methods can be categorized into two basic types lossless or lossy. Lossless compression methods represent the signal with a few bits while providing the same shape as the original speech signal at the decoder end, the run length encoding and the Huffman coding are the most known in this type. Lossy compression methods generate an inaudible distortion in the reconstructed signal. The possible compression ratio using lossy compression

is often much higher than by lossless methods [1]. Many standards of compression use both of them in order to increase the compression ratio. For example, the MPEG-Layer3 uses a lossy compression and Huffman coding. There are three key speech compression techniques, namely the waveform coding, parameter extraction and transformation methods. The waveform coding consists of removing correlation between speech samples to reduce the bit rate. It aims to minimize the error between the reconstructed and the original speech signal. Waveform coding schemes almost have low complexity, whereas their compression factors are also low. The simplest form of waveform coding is Pulse Code Modulation (PCM) which is defined in the ITU-T G.711 specification. The parameter extraction method is inspired from speech mechanism. It extracts the features of the signal which are then coded into binary bit stream. Compared to waveform-based codec, parametric-based codec has a high complexity, but can achieve a better compression factor. A typical parametric codec is Linear Prediction Coding (LPC) [2] [3]. The third technique is the transform based compression. It converts the signal from the time domain to another parsimonious domain. Many mathematical transforms have been exploited for audio compression. (E.g., discrete cosine transform, wavelet transform...). Among them, the wavelet transform is the most popular one since it was used in many coders [4] [5] [6] [7] [8]. The basic principle of Discrete Wavelet Transform (DWT) consists in separating the signal into two sets, one representing the general shape of the signal and the other representing its details. The general shape of a signal is represented by its low frequencies and the details are represented by its high frequencies. In order to make the separation between them, a pair of filters is needed: a low-pass filter that extracts the general shape called approximation, and a high-pass filter which estimates its details. The output of the high-pass and low-pass filters are down sampled by a factor of two.

Previous studies have proven that wavelet outperforms the FFT (fast Fourier transforms), DCT (Discrete Cosine Transform.) [12], and LPC (Linear Predictive Coding) [13].

In recent years many researches in the field of digital signal processing show interests on Discrete Hartley Transform (DHT) [14] [15][16]. Computing the DHT directly from its definition is too slow which does not fit with real-time application in which the computational time has a great importance. In above context, we have proposed a real-time speech compression system based on Fast Hartley Transform

(FHT). We have also proposed a modified scheme for Run Length Encoding (RLE) to improve compression factor.

The content of this paper is structured as follows; the next section shows the mathematical formulation for FHT. Section III describes the different stages of the proposed algorithm followed by the evaluation criteria. Section V exposes the simulation results. The real-time implementation is detailed and evaluated in section VI. Finally, we conclude this work in section VII.

II. ALGORITHM FORMULATION FOR FHT

The Fast Hartley Transform uses two properties of DHT to reduce the number of computations. The first property is that the kernel for Hartley Transform is periodic. The second is that temporal shifting correspond to multiplication in the frequency domain. The generalized Discrete Hartley Transform (DHT) is defined for sequence $x(n)$ by the following equation:

$$H(k) = \sum_{n=0}^N x(n) \text{cas}\left(\frac{2\pi kn}{N}\right) \quad (1)$$

And the inverse transformation can be defined as:

$$x(n) = \frac{1}{N} \sum_{k=0}^N H(k) \text{cas}\left(\frac{2\pi kn}{N}\right) \quad (2)$$

Where $\text{cas}(x) = \cos(x) + \sin(x)$. From the above equations it can be observed that both forward and inverse transformations can be computed in the same kernel except the constant multiplication. Eqn. (1) can also be decomposed as:

$$H(k) = \sum_{n=0}^{\left(\frac{N}{2}\right)-1} x(n) \text{cas}\left(\frac{2\pi kn}{N}\right) + \sum_{n=N/2}^{N-1} x(n) \text{cas}\left(\frac{2\pi kn}{N}\right) \quad (3)$$

Let $n = n + N/2$ in the second summation of Eqn. (3).

$$H(k) = \sum_{n=0}^{\left(\frac{N}{2}\right)-1} \{x(n) \text{cas}\left(\frac{2\pi kn}{N}\right) + x\left(n + \frac{N}{2}\right) \text{cas}\left(\frac{2\pi k\left[n + \frac{N}{2}\right]}{N}\right)\} \quad (4)$$

Separating even and odd part of the input points, and representing $H(2k)$ and $H(2k + 1)$ for even and odd respectively, eqn.4 may be rewritten as:

$$H(2k) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x\left(\frac{N}{2} + n\right)] \text{cas}\left(\frac{4\pi nk}{N}\right) \quad (5)$$

$$H(2k + 1) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) - x\left(\frac{N}{2} + n\right)] \text{cas}\left(\frac{2\pi n(2k+1)}{N}\right) \quad (6)$$

A more complete development of the FHT can be found in [13]. The DHT requires N^2 multiplication whereas the FHT requires only $N \log_2 N$ multiplication.

III. THE PROPOSED ALGORITHM

The block diagram of the proposed compression system is illustrated in Fig. 1. The different steps of the system are explained in the succeeding paragraphs.

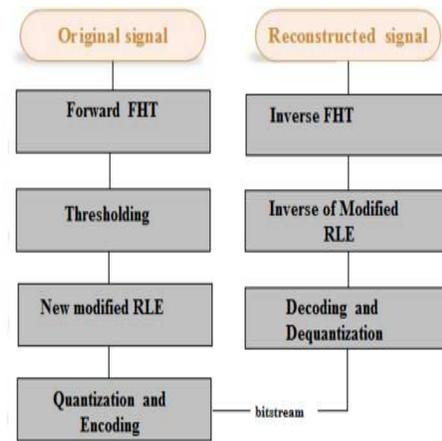


Fig. 1. Block diagram of a the proposed speech

A. Fast Hartley transform

The first step of our approach consists in decomposing the speech signal using FHT. It converts the temporal representation of a signal into a frequency representation. This domain transformation reduces the redundancy and decorrelates the signals samples, so, decreases the bitrates of transmission. The FHT concentrates speech information into a few coefficients as shown in Fig. 2. Therefore, after analyzing the Hartley transform of a signal, many coefficients will either be zero or have negligible magnitudes.

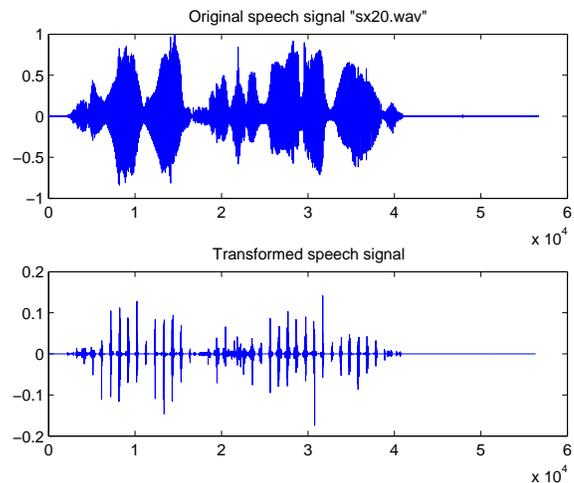


Fig. 2. Normalized hartley coefficients of speech signal(sx20.wav)

B. Thresholding

Thresholding is the most important step in a transform based compression; it consists of rejecting the coefficients of the FHT transform inferior to a given threshold. There are different methods of thresholding, such as the hard and the soft thresholding which are the commonly used methods. In this work we have used the hard thresholding given by this equation:

$$C_{Re} = \begin{cases} C_{Re} & \text{if } |C_{Re}| \geq T \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

C. Modified Run Length Encoding

The majority techniques based on transform coding use the zero run length encoding (ZRLE). Indeed, the thresholding increases the number of consecutive zeros. ZRLE is a very simple method of data compression in which a run of zeros is encoded using two values. The first value indicates the start of the sequence while the second value indicates the number of zeros in this sequence. For example, the sequence 1,0,0,0,0,2,0,0,0,0,0 would be encoded as 1, 0, 5, 2, 0, 6. Compression factor (CR) for this example is 2.6.

As we can see from the above example, ZRLE works better with data where there are successive runs of zeros. If the number of zeros in each run is longer than two, compression factor will be greater than or equal to 1. Whereas, many single zeros in data make expansion of data instead of compression. In order to overcome this problem and to increase the compression factor, we have proposed a modified scheme of run length encoding which takes into account that the speech signal is normalized to the range [-1,1] .

It consists first of replacing the samples of amplitude 1 by 0.99 and -1 by -0.99. Then, we code the each run of zeros by a one value which represent at the same time the start of the sequence and the number of zeros. The proposed modified run length encoding overcomes the problems of the classic ZRLE and increases the compression factor. **Example1:**

Input = {0.6, 0.2, 0.8, 0, 0, 0, 0.9, 0}.

Output with MRLE = {0.6, 0.2, 0.8, 4, 0.9, 1}: CR=1.3.

Output with ZRLE = {0.6, 0.2, 0.8, 0, 4, 0.9, 0, 1}: CR=1.

Example2:

Input= {0, 0.2, 0.3, 0, 0.8, 0.3, 0, 0.5, 0}.

Output with MRLE= {1, 0.2, 0.3, 1, 0.8, 0.3, 1, 0.5, 1}: CR=1.

Output with ZRLE= {0, 1, 0.2, 0.3, 0.1, 0.8, 0.3, 0, 1, 0.5, 0, 1}: CR=0.75(expansion).

Exemple3:

Input= {0, 0,0,0,0.2, 0.3, 0, 0, 0, 0}.

Output with MRLE= {4, 0.2, 0.3, 4}: CR=2.5.

Output with RLE= {0, 4, 0.2, 0.3, 0, 4}: CR =1.66. The signal is reconstructed by checking the type of each sample. Thus, the integer type of data informs about the number of zeros whereas the non-integer values present necessarily the samples of the signal.

IV. EVALUATION CRITERIA

The metrics used for the assessment of the quality of reconstructed signal are either objective or subjective. Objective evaluation criteria are based on mathematical parameters which do not require much material and time consumption. The majority of these criteria are defined in the temporal domains. Some frequently used metrics are listed below.

- Compression ratio (CR)

$$C = \frac{\text{length}(x(n))}{\text{length}(c(n))} \quad (8)$$

$c(n)$,is the compressed signal.

- Signal to noise ratio (SNR)

$$SNR = 10 \log_{10} \left(\frac{\delta_x^2}{\delta_e^2} \right) \quad (9)$$

Where, δ_x^2 is the mean square of the speech signal and δ_e^2 is the mean square difference between the original and reconstructed signals.

- Peak Signal to noise ratio (PSNR)

$$PSNR = 10 \log_{10} \frac{NX^2}{|x-r|^2} \quad (10)$$

N ,is the length of the reconstructed signal, X , is the maximum absolute square value of the signal and $|x-r|^2$ is the energy of error between the reconstructed and original signal.

- Normalized root mean square error (NRMSE)

$$NRMSE = \sqrt{\frac{(x(n) - r(n))^2}{(x(n) - \mu_x(n))^2}} \quad (11)$$

$x(n)$, is the speech signal, $r(n)$ is the reconstructed signal, and $\mu_x(n)^2$ is the mean of the speech signal.

- Absolute Category Rating

Several methods for subjective assessment are used in literature which are described in ITU-T Recommendation P.830. The most commonly used evaluation method is the absolute category rating (ACR) in which a group of listeners listen to audio sequences and then judge the perceived quality according to a rating scale. The ACR is frequently used in ITU-T applications like G711, G728, G711. The average numeric score over all experiments provides a score called Mean Opinion Score (MOS). We present in the table below the correspondence between the scores and the different quality judgments.

TABLE I. RATING SCALE AND DESCRIPTIONS FOR THE ABSOLUTE CATEGORY-RATING (ACR) TEST

Rating	ACR description-MOS
5	Excellent
4	Good
3	Fair
2	poor
1	Bad

V. TEST AND RESULTS

In this section, a Matlab program has been developed to implement the speech compression codec based on FHT with MRLE. To evaluate the efficiency of the developed algorithm, a comparative study between the wavelet based compression and the proposed system is performed using objective criteria; CR, SNR, PSNR and NRMSE. In all simulations, only source speech signals extracted from the TIMIT database are exploited [14].

Throughout the figures below, it is observed that the proposed system rates are better than those obtained by DWT based compression. Fig.6 reveals the superiority of the proposed algorithm. In fact, it gives the lowest NRMSE. The results of SNR and PSNR show a gain of 2db compared to the wavelet based compression. The compression factor reached by the proposed algorithm is increased from 2.5 to 7. To prove the reliability of the proposed system in preserving the speech quality, we have compared the objective criteria (SNR, PSNR and NRMSE) obtained by applying FHT and the adopted algorithm. So, we have remark that the criteria are maintained the same which demonstrates that the proposed system does not affect the signal quality.

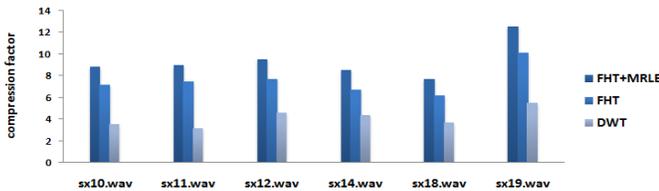


Fig. 3. Compression Ratio (CR) variation using FHT, MFHT and DWT algorithms

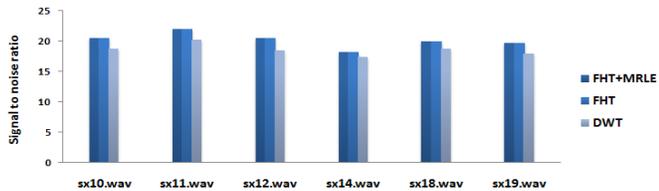


Fig. 4. Signal to noise ratio variation using FHT, MFHT and DWT algorithms

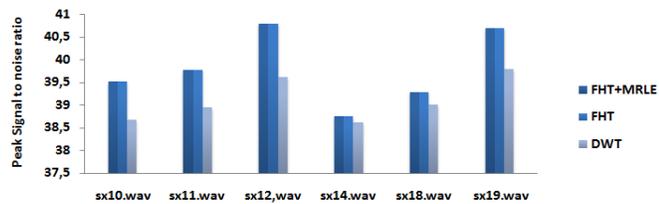


Fig. 5. Peak to noise ratio variation using FHT, MFHT and DWT algorithms

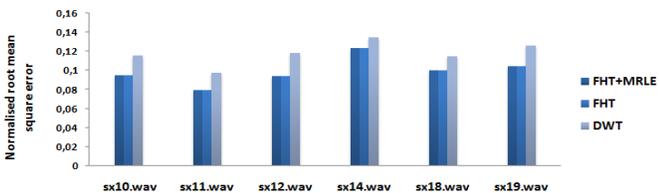


Fig. 6. Normalized mean square error variation using FHT, MFHT and DWT algorithms

VI. REAL-TIME IMPLEMENTATION BASED ON DSP

Real-time test has a great importance, especially for audio applications which have a strict timing constraints such as

audio streaming. In real-time application, the input signal and the generated output can be processed continuously that explains that the mean processing time per sample is lower than the sampling period. So, we have tested our algorithm on a flexible platform which convenient with the particularity of our application. For this purpose we have used a developed starter kit containing a DSK board based on DSP-TMS320C6416 and the software tool (Code Composer studio). We have also used a rapid prototyping tool from Mathworks.

A. DSK C6416 Overview

The DSK C6416 board includes a fixed point digital signal processor TMS320C6416 which operates with a clock frequency of 1GHz. It is also equipped with audio codec TLV320AIC23 (AIC23) that provides analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) functions with a selecting sampling rate ranged of alternative settings from 8 to 96 kHz. As indicated in the figure below the DSK board has four connections which provide analog inputs and outputs: A microphone input port, a line in port, a line out port, and a headphone port. The DSK board includes 16 MB (megabytes) of synchronous dynamic RAM (SDRAM) and 512 kB (kilobytes) of flash memory. The TMS320C6416 is based on the very long instruction word (VLIW) architecture, which is well suited for numerical intensive algorithms. The internal program memory is structured so that a total of eight instructions can be fetched every cycle. For example, with a clock rate of 1 GHz, the C6416 is capable of fetching eight 32 bit instructions every 1/ (1 GHz) or 1.0 ns. Fig. 7 presents an overview of the Spectrum Digital DSK board and the AIC23 Codec.

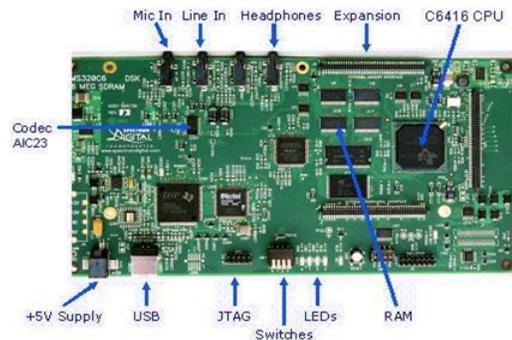


Fig. 7. TMS320C6416 DSK development board from Texas instruments

B. Rapid Prototyping Technology

For a rapid prototyping of real-time applications on DSP processor, we have used the Embedded Target for C6000 DSPs Platform and RTW (Real-time Workshop). They convert the Simulink model into efficient C code specifically for C6000 processors. MATLAB Link for CCS Development Tools interface with TI CCS (code composer studio) to produce an executable that is loaded into the DSK-C6416. Fig. 8 shows the general scheme for communication between Matlab \ Simulink and the embedded DSP target;

Fig.9, shows the designed Simulink model for implementing the audio codec based on FHT and on TMS320C6416. The



Fig. 8. Flow Diagram connecting Simulink and Real Time Workshop with DSK C6416

DSK-C6416 block has a prime importance. Indeed it provides access to the processor hardware settings. It is extracted from the C6000 Target Preferences library.

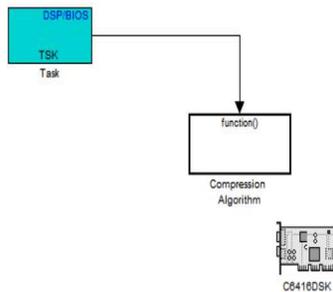


Fig. 9. Simulink model of audio compression system on TMS320C6416

Fig.10 shows the Simulink blocks of FHT, which is executed using a TSK thread of DSP/BIOS. This latter is a scalable real-time kernel designed to be used by applications that require a real-time scheduling and synchronization. It provides preemptive multi-threading, hardware abstraction, real-time analysis, and configuration tools. It is also designed to minimize CPU requirements and memory on the target [15]. The Simulink block "compression algorithm" presents a subsystem that contains the different stages of the compression algorithm.

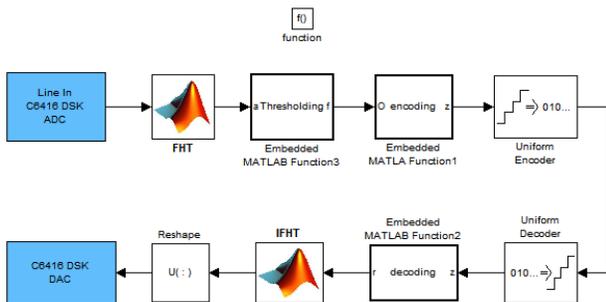


Fig. 10. Simulink model of the proposed audio codec

In order to validate the performances of the proposed speech compression algorithm, we have calculated the number of cycles and the memory consumption. Table II presents the

number of cycles in MCPS and the memory consumption in kilobytes (KB) required to running the audio compression system. From the above table, we can remark that total MCPS re-

TABLE II. CYCLE COUNT AND MEMORY CONSUMPTION OF THE PROPOSED SPEECH COMPRESSION ALGORITHM

Algorithm	Total (MCPS)	Memory consumption (Ko)
Compression with DSP /BIOS	5.033	234
Without DSP/BIOS	37.146	200

quired for running the FHT code decreases by using DSP/BIOS tool. The CPU speech and the memory consumption meet the real time processing requirements of TMS320C6416 (CPU speed = 1 GHz, 512KB of flash memory and 16MB of SRAM). Compared to the results obtained in [16], which highlights a real-time implementation of speech compression using wavelet, our work brings some improvements in term of complexity. Due to the accuracy of subjective evaluation, we have considered the listening test ACR. 10 volunteers have evaluated the quality of the sentences generated from the headphone connected on the DSP board. The tested sentences are pronounced with different dialects (English, French and Arabic). The mean opinion scores (MOS) resulted from the listening ACR is 4 which is close to the MOS of the pulse code modulation (PCM). From this test we have noticed that the change of dialect does not affect the quality of our algorithm.

VII. CONCLUSION

In this paper, a new speech compression algorithm using fast Hartley transform combined with a modified run length encoding scheme has been presented. The proposed algorithm proves its reliability to improve the speech compression ratio without affecting the signal quality referring to the performance evaluation made using the objective criteria such as: CR, SNR, PSNR and NRMSE. In this context, a comparative study with the wavelet based compression has demonstrated that the application of our algorithm increases the compression factor from 2,5 to 7 without sacrificing neither the speech intelligibility nor the quality. Finally, the real-time test of speech compression codec has been successfully implemented in TMS320C6416 platform and reveals that the proposed algorithm has significantly decreased the system complexity mostly when DSP/BIOS is intervening.

As a future work, we tend to integrate a voice activity detection (VAD) to improve the coder performances.

REFERENCES

- [1] K.Sayood, "Introduction to Data Compression, Third Edition", Morgan Kaufmann Publishers, 2006.
- [2] M.R. Madane, Z. Shah, R. Shah, S. Thakur, "Speech Compression Using Linear predictive Coding", Proceeding of the international workshop on machine Intelligence Research (MIR Day GHRCE- Nagpur) 2009 MIR Labs
- [3] P. Venkateswaran, A. Sanyal, S. Das, R. Nandi, S. K. Sanyal, "An Efficient Time Domain Speech Compression Algorithm Based On LPC and Sub-Band Coding Techniques", Journal of Communication, vol. 4, no. 6, July 2009.
- [4] Aloui, Noureddine, Chafik Barnoussi, and Adnane Cherif. "New algorithm for QMF banks design and its application in speech compression using DWT" *Int. Arab J. Inf. Technol.* 12.1 (2015): 86-93.

- [5] A.Noureddine, S. Bousselmi, and A. Cherif. "Optimized Speech Compression Algorithm Based on Wavelets Techniques and its Real Time Implementation on DSP", *International Journal of Information Technology and Computer Science (IJITCS)* 7.3 (2015).
- [6] Kumar A, Singh GK, Rajesh G, Ranjeet K,"The optimized wavelet filters for speech compression" ., *International Journal of Speech Technology*. 2013 Jun 1;16(2):171-9.
- [7] Khalifa, Othman Omran, Sering Habib Harding, and Aisha-Hassan Abdalla Hashim. "Compression using wavelet transform", *International Journal of Signal Processing* 2.5 (2008): 17-26.
- [8] Elaydi, Hatem, Mustafi I. Jaber, and Mohammed B. Tanboursa. "Speech compression using wavelets",*The International Arab Conference on Information Technology*. 2003.
- [9] Rajesh, G., A. Kumar, and K. Ranjeet. "Speech compression using different transform techniques", *Computer and Communication Technology (ICCCCT)*, 2011 2nd International Conference on IEEE, 2011.
- [10] Najih AM, Ramli AR, Ibrahim A, Syed AR. "Comparing speech compression using wavelets with other speech compression schemes", *In Research and Development*, 2003. SCORED 2003. Proceedings. Student Conference on 2003 Aug 25 (pp. 55-58). IEEE.
- [11] A. Fatima and N. K. Nishchal, "Image encryption using Hartley transform domain polarization encoding" in *13th International Conference on Fiber Optics and Photonics*, OSA Technical Digest (online) (Optical Society of America, 2016), paper Tu4A.27.
- [12] K. C. Narendra and S. Satyanarayana, "Hartley transform based correlation filters for face recognition" *2016 International Conference on Signal Processing and Communications (SPCOM)*, Bangalore, 2016, pp. 1-5.4
- [13] Poularikas, Alexander D., ed. "Transforms and applications handbook", *CRC press*, 2010.
- [14] Zue V, Seneff S, Glass J "Speech database development at MIT: TIMIT and beyond", *Speech Communication*. 1990 Aug 1;9(4):351-6.
- [15] Texas instruments, "TMS320 DSP/BIOS v5. 42 user s guide ", 2010-01-20 .
- [16] Aloui, N., Bousselmi, S., Cherif, A. (2015) "Optimized Speech Compression Algorithm Based on Wavelets Techniques and its Real Time Implementation on DSP"*International Journal of Information Technology and Computer Science (IJITCS)*, 7(3), 33.