Higher Compression Rates for GSM 6.10 Standard Using Lossless Compression

معدل ضغط أعلى لبيانات معيار(جي إس إم GSM 6.10) باستخدام الضغط الأقل فاقدية

Islam Younis Amro
Associate Professor/ Al-Quds Open University / Palestine
iamro@qou.edu

اسلام يونس عمرو
أستاذ مشارك / جامعة القدس المفتوحة / فلسطين

Received: 18/10/2020, Accepted: 04/09/2021
DOI: https://doi.org/10.33977/2106-000-005-001
https://journals.qou.edu/index.php/PJTAS

تاريخ الاستلام: 2020/10/18، تاريخ القبول: 2021/09/04
E-ISSN: 2521-411X
P-ISSN: 2520-7431

Abstract

This research aims at exploiting the lossless Hamming correction code compression algorithm (HCDC) to reduce the transmission data rate in the GSM 6.10 standard, which holds several similarities with modern adaptive multi-rate codec in coefficients calculations and excitation principles. The compression algorithms depend on the properties of the hamming codes where data bits can be calculated from the parity bits. In this research, we chose parity equals 3 and data bits equals 4. Several iterations were conducted over the compressed frame information to achieve even higher compression rates. The compression rate was implemented over the standard of GSM 6.10, which is a variation Code Exited Linear Prediction coding (CELP). Regarding the data samples selected to conduct the test, two males and two females' voice file samples at 8khz and quantized on 8-bit resolution were selected. The duration of the files varies from 4 to 6 seconds. Each sample was divided into 20ms frames; each frame was expressed using GSM6.10 with 260 bits of data included Linear perdition coefficients, pitch period, gain, peak magnitude value, grid position, and the sample amplitude. This shows that the 260 bits every 20ms form a data rate of 13kbps. The 260 bits were subjected to HCDC, and the data rate was reduced by 60%, reaching down to 5kbps on average. The results compared to the famous FLAC lossless audio compression, which showed 15% compression only. The research did not consider any quality testing since the compression is lossless. The research used standard ITU libraries to conduct the GSM6.10 data acquisition and open-source platforms for FLAC.

***Keywords**: Linear prediction coding, lossless compression, speech compression, source coding, cellular communication.*

الملخص

يهدف هذا البحث إلى توظيف خوارزمية ضغط البيانات المعتمدة على ترميز (هامينج) للتصحيح (HCDC) دون فقدان البيانات؛ وذلك لتقليل معدل بيانات الإرسال لمعيار (جي إس إم GSM 6.10) والذي يتم استخدام أسس الترميز الخاصة به في معظم الترميزات الحديثة من احتساب معاملات خطية، وطرق احتساب إشارة تفعيل الفلاتر الخطية بشكل عام، تعتمد خوارزمية الضغط المذكورة على توظيف خصائص ترميز (هامينج) بحيث يتم استخدام (البت) الخاص بحقل البيانات كأساس في احتساب (البت) الخاص (بالباريتي)، وعليه يتم ارسال (البتات) الخاصة بالبيانات، و توفير (البتات) الخاصة (بالباريتي). في هذا البحث، قمنا باعتماد مجموعة من (البتات) تتكون من سبعة حقول بحيث يكون عدد (البتات) الخاصة بالبيانات (4)، و عدد (البتات) الخاصة (بالباريتي) (3)، و تم أيضا إعادة ضغط البينات المضغوطة مرات عدة متتالية للحصول على معدل ضغط أعلى. تم تطبيق عملية الضغط على المعيار (GSM6.10) والذي يعتبر أحد أنواع ترميز التوقع الخطي المحفَّز خارجيا، تم استخدام عينات من أصوات ذكرين بالغين وانثيين بالغتين، وأخذت عينات الإشارة بتردد (8) كيلو هيرتز ورقمتها بثماني بتات، مدة كل ملف عينة من أربع الى ست ثوانٍ، وتمّ تقطيع كل عينة إلى نوافذ طول النافذة عشرون ميلي ثانية، ومن ثم ترميز كل نافذة باستخدام (GSM6.10)، والتي بدورها تنتج (260) بتا كل عشرين ثانية. تشتمل هذه (البتات) على: معاملات توقع الخطة، تردد الصوتي الذبذبة (Pitch)، التحصيل (Gain)، قيمة القمة الأعلى للإشارة، موقع الشبكة (Grid position)، قيمة العينة. يتم تجمي ال (260) بتا، ومن ثم يتم ضغطهم مرات عدة باستخدام خوارزمية HCDC. تم تخفيض معدل الارسال من (13) كيلوبت في الثانية إلى قرابة (5) كيلوبت بالثانية في المعدل. تمت مقارنة هذه النتائج بخوارزمية (FLAC)، والتي حققت نسبة ضغط بمعدل (15%) فقط. وبما أن الضغط المستخدم هو ضغط لا يفضي إلى فقد البيانات (Lossless)، لم يتم التطرق إلى دراسة جودة الإشارة في هذه البحث. وتم استخدام المكتبة القياسية لترميز (GSM6.10)، والمتوفرة على موقع اتحاد الاتصالات (الدولب) إلى جانب مكتبة (FLAC) مفتوحة المصدر.

الكلمات المفتاحية: ضغط البيانات، ضغط الصوت، الشبكات الخليوية، معاملات التوقع الخطي، الترميز المصدري، ضغط بيانات بلا خسائر.

INTRODUCTION

Audio and speech compression might be considered the most diverse aspect in the data compression discipline. This is due to the diversity of its domains, data representation methods, and the high demand for high quality and lower data rate paradox. Not to forget, the complexity constraints over any algorithm are to be proposed (Wu et al. 2002).

The basic form of any signal is acquired after its quantization (Openhaim, 1997). This is the point where all digital compression algorithms start; a well-known followed track is the linear prediction coding compression approach due to its low rate, good quality, and acceptable complexity. It became the hardcore of modern voice communication systems (Kain et al., 2001; Wah, 2005) and the raw data form for artificial intelligence applications on speech. (Wu et al., 2002, Lam et al., 2000).

As for the review of lossless audio compression standards and algorithms proposed by AbdulMuin et al. (2017); it shows that several compression approaches are used either in row PCM form or in other coding formats, mainly based on Huffman methods. Recent implantation was found in the study of Uttam, 2019, achieving lossless compression of audio by encoding its constituted components (LCAEC), which are based on Huffman and Burrows–Wheeler transform. On lossless audio compression based on heuristic methods based on neural networks found in Uttam, 2019, several hidden layers have been implemented in the proposed network for the present encoding framework based on deep learning process. Another lossless audio compression method is incorporated by the nature of channels of transmission and the types of data like in Takehiro et al., 2019, where the compression is considered in terms of video compression channel and is based on MPEG multichannel audio compression. Another statistical compression method found in Yanzhen et al., 2019. This method is based on pulse destitution modeling then generates a fixed codebook that enables AMR features. For spatial audio decoding and compression, extensive research was conducted by Menzies et al., 2017. The research considered decoding and compressing channel and scene objects to reduce processing complexity. In Luo et al., 2017, an auto encoder was exploited to detect the double compression for AMR. This research was useful in detecting several compressions for the same block when several transmission rates are used. Another research on statistical methods of auditory representation was found in Biesmans et al., 2017. Based on canonical correlation analysis, that emulates the auditory system signaling in EEG, brain is stimulated directly by passing the human auditory system. The importance of this research lies in how to generate and EEG signal from an audio signal. This is a new form of coding and compression.

This research exploits a new lossless compression algorithm based on the Hamming Correction Code Compression (HCDC) explained in Bahadili, 2007, in compressing speech/audio signals in its GSM 6.10 form. Similar work was conducted in Amro et al., 2011, using this compression algorithm over, and an experimental vocoder that exploits residual signal as excitation using Discrete Cosine Transform (DCT) with considerable compression ratio. The compression algorithm in this research addressed the linear prediction coefficients only without addressing the DCT excitation signal. The HCDC algorithm was also exploited in compressing audio signal based on Code excited linear prediction coding in Amro, 2013. In this research, both the excitation signal and linear prediction coefficients were addressed and achieved a good average compression rate.

Although several GSM 6.10 standards were promoted to Adaptive Multirate (AMR) codec, to enhance quality, in addition to the Adaptive Multirate Narrow-Band (AMR-NB) codec, which works in the telephony bandwidth in addition to the Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) systems. The founding principles of the GSM 6.10 are available in these codecs, such as the linear prediction coefficients calculation approach based on the Levinson Durban recursion and the quantization of the excitation signal, in addition to gaining value. These parameters are present in all generations of codecs and exploited in cellular communication (GSM 2020), making it easier for this research to prove the concept over GSM 6.10 with the possibility to generalize the results of the scale of different rates in the AMR in the future.

The following section discusses the GSM 6.10 encoding and decoding. We elaborate on the properties of the algorithm exploited and then mention the methodology and experiment design in the following section. The results are presented in the following section with comments and analysis. Then we finalize with a summary and conclusion.

The GSM 06.10 full rate

THE GSM 06.10 full rate coder is considered a hybrid code which is a form between waveform coders and vocoders. Waveforms coders consider the processing among physical characteristics of the signal in the time domain, frequency domain, or any other transfer function domain. Vocoders have their own domain based on linear prediction coding (LPC). LPC works on the classification of speech signal as voiced or unvoiced. A voiced signal is formed from sound in which certain turbulence happens in vocal cords. This turbulence has a certain frequency which is called pitch. A pitch is a train on impulse with known frequency and gain, represented in the linear prediction domain. The frequency of this signal in the LPC domain is the pitch frequency for a given voice. The unvoiced signals in the LPC are incorporated with voices that do not include vocal tracts turbulence, like the letter S. This kind of letter has no certain frequency in the LPC domain. Thus, it is expressed as white noise. Both white noises and/or the pitch impulse train are synthesized with digital filer with certain order (10 minimum and usually 12). The filter is the linear synthesis digital filer, and its coefficients are calculated from time-domain parameters from the signal. This process synthesizes the spoken voice back. The quality of the output signal in terms of physical signal qualities (objective), such as signal-to-noise ratio (SNR) and Segmental Signal to noise ratio (SSNR), is considered low. However, it still can be heard and understood. That is why a special qualitative (subjective) technique is adapted based on voting. This quality assessment method is known as the Mean of Score (MOS), and it usually ranges from 0 to 5. However, 3.5 is the range of good and acceptable quality (Chu, 2003).

The GSM uses a compression approach that utilizes both waveform methods and LPC Methods. In GSM speech encoder, the encoder takes 13 bits as input as Pulse Code Modulation (PCM) signal from audio part of a mobile station or from the network or Public Switched Telephone Network (PSTN) via an 8bit / A-law to 13 (13bit* 8KHz=104Kbps) bit uniform PCM (Malvar, 2007). The encoded speech output is delivered to a channel encoder unit specified in GSM 05.03 (Hu et al., 2007).

On the receiving side, an inverted operation takes place as described in GSM 06.10. The process is based on a mapping between inputting 160 speech samples, each is 13-bit uniform PCM, then it is exploited to encode 260 bits, and from encoded blocks of 260 bits to generate an output of 160 reconstructed speech blocks. The rates are 8K samples per second, generating an encoded bitstream of 13kbps. This coding scheme is known as regular pulse excitation long-term prediction linear predictive coding.

GSM Full Rate Encoder

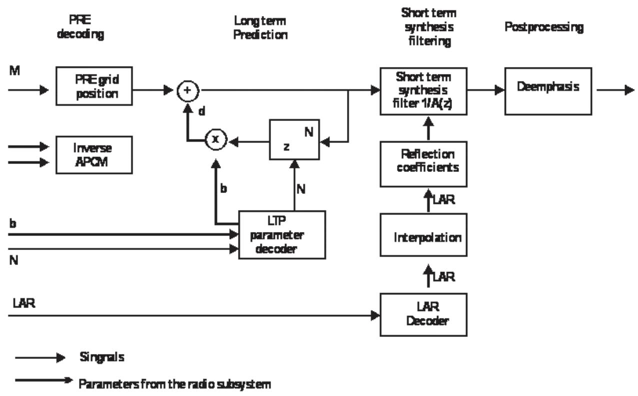

Figure 1 GSM Encoding

Figure 1 shows a detailed block diagram of GSM 06.10 Speech Encode. The speech input frame, made of 160 samples, is the first step to generate an offset-free signal, then a pre-emphasis filter is applied. Then160 samples were used to determine the short-term LPC coefficients through the LPC analysis. This process is conducted by calculating the Lavinoson Durban coefficient, then calculating the LPC residual signal for the short-term signal. Before transmission, the filter parameters, reflection coefficients, and gain are transferred to Log Area Ratios (LAR). The speech frames are then slitted into 4 sub-frames with 40 samples of short-term residual signal in each. Each sub-frame is processed as a block by the following functional components. Before processing sub-blocks of 40 short term residual samples, the parameters of the long term analysis filter, the Long Term Parameter (LTP), and the gain are estimated in the LTP analysis block, based on the current sub-block of the present and a stored sequence of the 120 previous short term residuals. Then by subtracting 40 estimates of the short-term residual signal from the short-term residual signal itself, where a block of 40 long-term residual signal samples is acquired. In the next stage, the block of 40 long-term residual signal is fed to the Regular Pulse Excitation (RPE) stage that performs a basic compression function analysis. Resulting from the RPE stage, the block of 40 input long-term residual signal samples is represented by one 4 sub-sequences candidates with 13 pulses each. The 13 RPE pulses are then encoded using Adaptive Pulse Code Modulation (APCM) with an estimation of sub-block amplitude which is transmitted to the decoder side as side information. The RPE values are also supplied to the local RPE decode-and-reconstruct module, which produces a block of 40 samples. These samples are the quantized versions of the long-term residual signal. Adding the quantized 40 samples of the long-term residual to the blocks of short-term residual signal previously encountered, the reconstructed short-term residual signal is acquired. The block containing the short-term residual signal is consequently obtained. Then the reconstructed signal is inputted in the analysis filter, which produces a new block of forty short-term residual signal estimates. These estimates are forwarded to the next sub-block to complete the feedback loop (ETSI, 2010).

GSM Full Rate Decoder

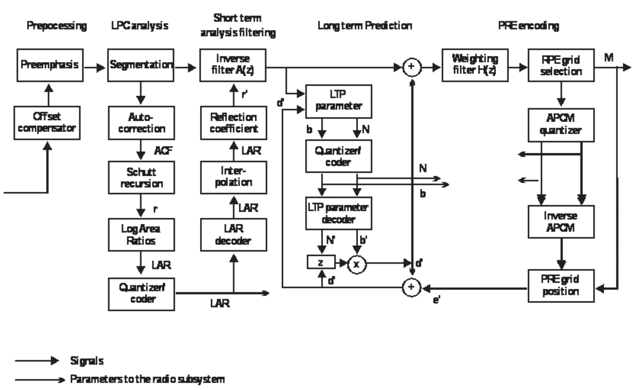

Figure 2 GSM Decoding

The GSM 06.10 Speech Decoder is shown in figure 2. As it can be seen, it includes similar stages to the feedback loop in the encoder. To ensure a zero-error transmission, the output must be the reconstructed short-term residual signal samples. These samples are inputted into a short-term synthesis filter. The next stage is the de-emphases filter in order to reconstruct the required speech signal. The GSM elaborated extensively on mapping input blocks of 160 samples in the original 13-bit uniform pulse code modulation format. This is done to encode 260 bits of blocks from encoded blocks of 260 bits of output blocks. This is obtained from 160 reconstructed speech samples. The average bit rate for the encoded stream is 13kbps obtained from 8000 samples per second. The bit allocation for the GSM full rate speech coding is seen in the table below and will be subjected to further compression using HCDC Algorithm. The frame length that is subjected in the process in 20 milliseconds.

Table 1 Bit allocation for GSM Full Rate Speech Coder (ETSI, 2010)

Parameter	No. per frame	Resolution	Total bits / frame
LPC	8	6,6,5,5,4,4,3,3	36
Pitch Period	4	7	28
Long Term Gain	4	2	8
Grid Position	4	2	8
Peak Magnitude	4	6	24
Sample Amplitude	4*13	3	156
Total			260

Hamming Correction Code Compression

Hamming Correction Code Compression (HCDC) is derived from hamming correction code. Let's consider the following set/ word of bits {*b0, b1, b2, b3, b4, b5, b6*}, re-expressing the set-in terms of its hamming version, we have {*p0, p1, d0, p2, d1, d2, d3*}, where number of parities=3 for a word of 7 bits length. In our research, we will transmit or save d bits only, and on the reception side, we will calculate the parity, so we can express 7 bits with 4 bits of data and save 3 bits. When we can do this process for the set of bits, we call it a valid word, which refers to the words' valid hamming calculation of data bits leads to the similar parity bits. If the word is invalid, this means that its data bit does not match its parity bits. In this case, we cannot compress it, and we have to transmit the word as is. We can compress valid words only; invalid words cannot be compressed since their actual bits don't match the ones calculated in hamming conditions. We mark valid words by 1 and invalid words by zero. This bit tells the decompressor what to do. In the case of a valid word, it means that we calculate the parity bits and place them in their right locations. In the case of an invalid word, we read 0 in the leading bit and read the whole word as is. Exhibited in Figure 3 is the compression algorithm, while Figure 4 exhibits the decompressor algorithm

1. *Initialization*
   - *Select* p
   - *Calculate* $n = 2^p - 1$
   - *Calculate* $d = n - p$
   - *Initialize* $b = 0$
2. *Read Binary Data*
   - *Read a Block of* n *bits length*
   - *[Add 1 to* b *]*
3. *Check block validity*
   - *If {Block = valid codeword} then*
     - *[Add 1 to* v *]*
     - *Write 1 followed by* d *block bits to the compressed file*
   - *Else {block= non-valid codeword}*
     - *[add 1 to* ω *]*
     - *Write 0 followed by* n *block bits to the compressed file*
   - *End if*
4. *if not end of data go to step 2*

Figure 3 HCDC Compressor

1. *Initialization*
   - *Select* p
   - *Calculate* $n = 2^p - 1$
   - *Calculate* $d = n - p$
   - *Initialize* $b = 0$
2. *Read Binary Data*
   - *Read the first bit* (h)
   - *[add 1 to* b *]*
3. *check for block validity*
   - *if {* $h = 1$ *}then*
     - *add [1 to* v *]*
     - *read* d *data bits*
     - *compute the hamming code for* d *write coded block to decompressed file*
   - *else {* $h = 0$ *}*
     - *[add 1 to* ω *]*
     - *Read block of* n *length*
     - *Write block* n *bits to he decompressed file*
   - *End if*
4. *if not end of data go to step 2*

Figure 4 HCDC deCompressor

Now we work on the evolution of its compression rate. The measuring references suggested in (Bahadili, 2008) are which represents the block size, i.e., the block we intend to analyze. The measuring references for Compression Rate suggested in the study of Bahadili, 2008 is, which represents the block size, the file to be compressed contains blocks, each is made of a number of bits, valid blocks count is expressed as and the invalid blocks are expressed as, the whole number of blocks can be expressed as:

$$b = \upsilon + \omega \quad (1)$$

This is a valid block led by 1 and an invalid one led by zero. So, the valid block is expressed by only its data bits excluding parity bits, the size of the valid block in the group is given by:

$$S_v = v(d+1) \quad (2)$$

For invalid blocks, the whole is used, so the size of the invalid blocks becomes

$$S_w = \omega(n+1) \quad (3)$$

And the size of the whole compressed file S_C becomes

$$S_c = nb + b - vp \quad (4)$$

The size of the compressed file in bits becomes

$$S_c = v(d+1) + \omega(n+1) \quad (5)$$

This can be written as

$$S_c = nb + b - vp \quad (6)$$

We know that the original file S_o is expressed as

$$C = \frac{nb}{nb+b-vp} \quad (7)$$

The compression ratio becomes

$$C = \frac{n}{n+1-rp} \quad (8)$$

expressing the ratio of valid blocks r as $r = \frac{v}{b}$.

The previous equation can be written as

$$C_k = \frac{S_o}{\prod_{i=1}^{k} C_i} \quad (9)$$

The algorithm can be iterated k times, where further compression can be achieved if the output of each phase is taken as an input for the next phase, the cumulative compression rate in this case the C_k, where k represents the number of iterations and C_i represent the compression on a given round, so if the code is to be compressed 8 times, then k is set to 8, and the compression rate becomes $\{C_1, C_2, \ldots, C_{7,} C_8\}$.

MATERIALS AND METHODS

The experiment was carried out on several data sets. Signals were S1(female), S2(female), S3(male), and S4(male). The samples S were for adult native English-speaking males and females. For each signal, we used 8-bit resolution at 8KHz sampling. The signal samples are segmented into a 20ms frame each, and the length of the samples ranges from 3 to 6 seconds each. The data to be compressed is obtained from Table 1 above, which includes the Bit allocation for the GSM Full Rate Speech Coder. For each 20ms of the sample signals, we will compress the 260 bits representing the GSM full rate speech coder information. The current data rate of the coder is 13Kbps. We will work on reducing this number in a lossless manner. In this case, there will be no need for quality detection. The performance of our work will then be compared for the FLAC algorithm since it is a wildly used lossless compression. The current transmission rate for the GSM algorithm used is 13Kbps obtained from 260 bits per sample over a window of 20ms. For each frame with data in table 1, we moved to the steps in figure 5, in order to evaluate the compression performance at parity =3 and at a different number of iterations. The selected number of iterations is 8 from practical experience. The iterations as mentioned above help enhance the compression ratio.

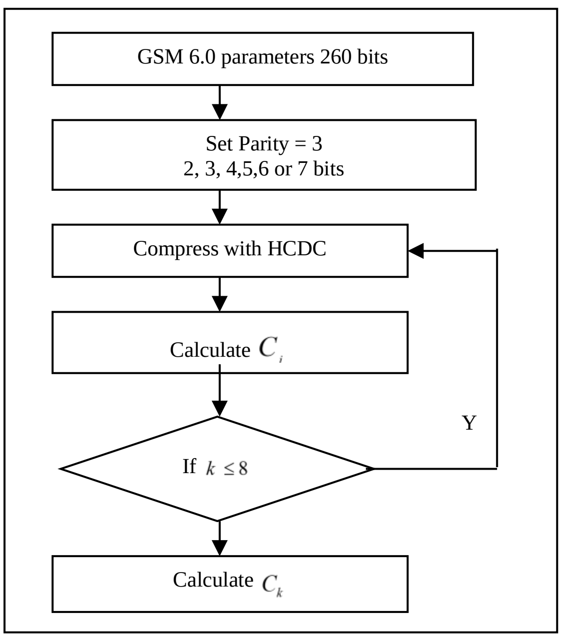

Figure 5 HCDC Experiment Design

The compression rate is to be calculated against the given parity = 3 on every count. The overall compression C_k is to be calculated for each sample accordingly and specified at the last iteration. The cumulative compression rate is then compared to the FLAC compression rate and the transmission rate. Then, the transmission rates are plotted together to see the average compression rate for the whole sample. This is calculated by averaging the rates for all frames within the sample for both HCDC and FLAC. Then performance notes are made.

RESULTS

For all the samples, compression was encountered only at parity=3. Table 2 below shows some of the best cases achieved with HCDC against frames at parity=3, the field Loop in the tables represents the compression turns, which iterates 8 times. The frame file size expressed the total number of bits in the frame. Valid Blokes represents the valid hamming codeword r as the valid blocks' ratio to the whole blocks in the file. Compression Ratio is C and computed by equation 8 above. Cumulative Compression is the and computed by equation 9, which represents the size rate between the original frame file and the current frame file size at the 8th iteration.

Table 2 One of the best cases achieved with HCDC against frames at parity=3

File Size	Total Block	Ratio	Valid	Invalid	Compressed	Ratio	Comm
60	37	0.46	17	20	228	1.14	1.14
228	32	0.5	16	16	192	1.19	1.36
192	27	0.44	12	15	168	1.14	1.55
168	24	0.46	11	13	148	1.14	1.76
148	21	0.43	9	12	132	1.12	1.98
132	18	0.44	8	10	112	1.18	2.33
112	16	0.31	5	11	108	1.04	2.42
108	15	0.27	4	11	104	1.04	2.51

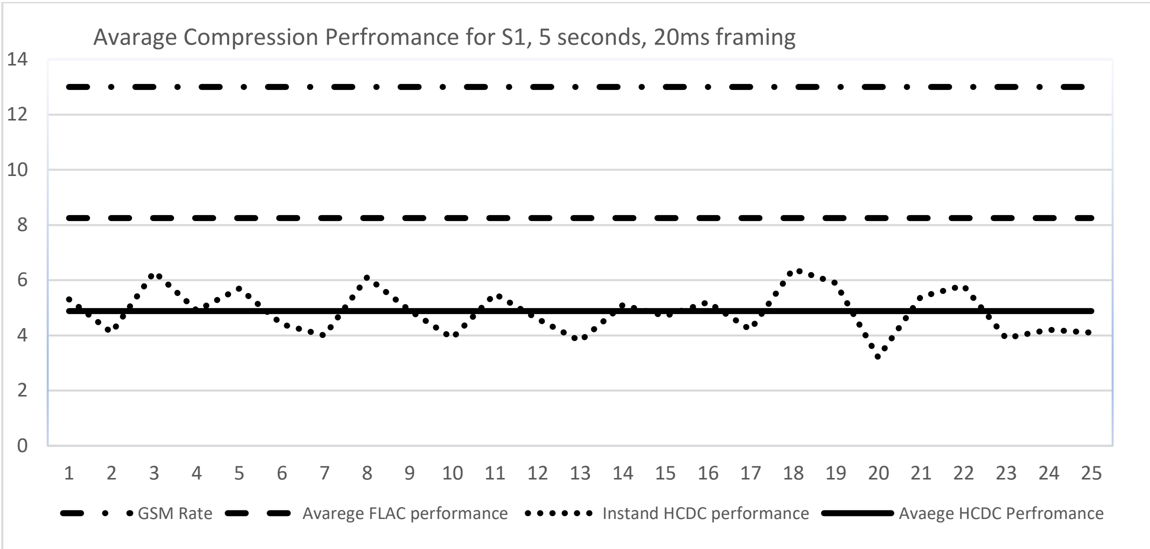

Figure 6 Average Transmission Rate for Sample Female 1

In Figure 6, we can see the algorithm has a very high potential of achieving lossless compression. The following table shows the frame information for the rest of the samples.

Table 3 Frame information for sample Male 2

File Size	Total Block	Ratio	Valid	Invalid	Compressed	Ratio	Comm
260	37	0.51	19	18	220	1.18	1.18
220	31	0.35	11	20	204	1.08	1.27
204	29	0.55	16	13	168	1.21	1.55
168	24	0.58	14	10	136	1.23	1.91
136	19	0.58	11	8	108	1.26	2.40
108	15	0.4	6	9	96	1.125	2.70
96	13	0.23	3	10	92	1.04	2.83
92	13	0.31	4	9	88	1.05	2.95

Table 4 Frame information for sample female 1

File Size	Total Block	Ratio	Valid	Invalid	Compresed	Ration	Comm
260	37	0.43	18	19	224	1.16	1.17
216	30	0.47	17	13	172	1.26	1.52
182	26	0.42	11	15	164	1.11	1.59
164	23	0.26	10	13	144	1.14	1.81
144	20	0.15	9	11	124	1.16	2.10
124	17	0.26	6	11	112	1.11	2.33
112	16		5	11	108	1.04	2.45
108	15		4	11	104	1.04	2.51

Table 5 Frame information for sample female 2

File Size	Total Block	Ratio	Valid	Invalid	Compresed	Ration	Comm
260	37	0.56	22	15	208	1.25	1.25
208	29	0.62	18	11	160	1.3	1.6
160	22	0.45	10	12	136	1.17	1.95
136	19	0.32	6	13	128	1.06	2.03
128	18	0.28	5	13	124	1.03	2.09
124	17	0.24	4	13	120	1.03	2.16
120	17	0.18	3	14	124	0.96	2.1
124	17	0.24	4	13	120	1.03	2.17

Average HCDC Compression performance and comparison

Table 6 Average performance of HCDC algorithm over given Samples

Sample	Duration in seconds	GSM kbps	FLAC Kbps	HCDC average Kbps	Reduction average (GSM to HCDC)
Male 1	5	13	8.25	4.86	62%
Male 2	6	13	7.89	5.21	59%
Female 2	5	13	9.14	4.79	63%
Female 2	6	13	9.21	4.88	62%

We can see from the table right above the general performance references regarding the HCDC compression. The FLAC has an average drop within 3 kbps. However, the challenge of FLAC since compression depends heavily on the nature of data. The file-based compression was used in this research, and the result was used as an average value in all of the cases. For the HCDC average Kbps, this is the average value of the broadcasted frames per file sample. As we can see, it achieved a very high compression rate with an average that exceeds 60% for all cases. In comparison to FLAC, it also achieved compression that exceeds FLAC but 40%. The HCDC is easier to implement and can give good performance in small blocks of data.

CONCLUSION

This paper exploits the Hamming Correction Code Compressor (HCDC) in compressing GSM full rate compression in a lossless manner. These parameters are calculated for every 20ms frame and then subjected to the lossless compressor. The parameters are the linear perdition coefficients, pitch period, gain, peak magnitude value, grid position, and sample amplitude. These parameters add up to 260 bits generated every 20ms. This information rate requires 13kbps to achieve the desired connection. This research implemented the HCDC compressor on the 260 every 20ms to achieve further lossless compression. We could reach data rates lower than 13kbps by 60%, reaching down to 5 kbps on average. The results were then compared to other lossless compression methods such as FLAC, and the algorithm we used showed better performance by 70% over FLAC. The research did not include any quality assessment due to the lossless nature of the algorithm.

References

- *AbdulMuin Fathiah, & Gunawan Teddy, & Kartiwi Mira, & Elsheikh A. (2017). A review of lossless audio compression standards and algorithms. Held in Malesia and Published in AIP Conference Proceedings 1883, 020006*
- *Amro Islam, & Abu Zitar Raed, & Bahadili Al-Bahadili (2011). Speech compression exploiting linear prediction coefficients codebook and hamming correction code algorithm. Springer International Journal of Speech Technology volume 14, Article number: 65*
- *Amro Islam (2013). Higher Compression Rates for Code Excited Linear Prediction Coding Using Lossless Compression. Presented in the Fifth International IEEE Conference on Computational Intelligence, Communication Systems and Networks. Madrid: Publisher IEEE*
- *Bahadili Hussein (2008). A novel lossless data compression scheme based on the error correcting Hamming codes. Elsevier international journal Computers and Mathematics with Applications. 56:143-14*
- *Bhatty P. Ninad, & Kosta P., & Kosta P. (2011). Proposed modifications in ETSI GSM 06.10 full rate speech codec and its overall evaluation of performance using MATLAB. International Journal of Speech Technology 14(3):157-165*
- *Biesmans Wouter, & Das Neetha, & Francart Tom, & Bertrand Alexander (2017). Auditory-Inspired Speech Envelope Extraction Methods for Improved EEG-Based Auditory Attention Detection in a Cocktail Party Scenario. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 25(5):402-412*
- *Chu H. (2003). Speech Coding Algorithms. New York: Wiley.*
- *European Telecommunications Standards Institute ETSI (2010). Digital cellular telecommunications system (Phase 2+) and Universal Mobile Telecommunications System (UMTS)and LTE and Network architecture V9.3.0. FRANCE:650 Route des Lucioles F-06921 Sophia Antipolis Cedex*
- *GSM Association (2020). Adaptive Multirate Wide Band Version 5.0. GSM Association Publications.*
- *HU YI, & Philipos C. Loizu (2007). Evaluation of Objective Quality Measures for Speech Enhancement. IEEE Transactions on Audio, Speech, and Language Processing 16(1):229-238*
- *Kain A., & Macon M. (2001). Design And Evaluation of A Voice Conversion Algorithm Based On Spectral Envelope Mapping And Residual Prediction. Centre For Spoken Language Understanding (CSLU), Oregon Graduate Institute, OR 97006, USA. 2000*
- *LAM Y., & Goodman J. (2000). A mathematical analysis of the DCT coefficient distributions for images. IEEE Transactions on Image Processing 9(10):1661-1666*
- *Lin Xiao, & Li Gang, & Li Zhengguo, & Chia Thien King, & Yoh Ai Ling (2001). A novel prediction scheme for lossless compression of audio waveform. Presented in IEEE International Conference on Multimedia and Expo: Publisher IEEE, Japan*
- *Luo Da, & Rui Yang, & Bin Li, & Jiwu Huang (2017). Detection of Double Compressed AMR Audio Using Stacked Autoencoder. IEEE Transactions on Information Forensics and Security. 12(2):432-444*
- *Malvar Henrique (2007). Lossless and Near-Lossless Audio Compression Using Integer-Reversible Modulated Lapped*

Transforms. Presented in Data Compression Conference DCC07: Publisher IEEE, USA

- *Menzies Dylan, & Filippo Maria Fazi (2017). Decoding and Compression of Channel and Scene Objects for Spatial Audio. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 25(11):2138-2151*
- *Openhaim et al. (1997). Signals and Systems. Second edition. NJ, USA: Prentice-Hall.*
- *Rao A.V., & Ahmadi S., & Linden J., & Gersho A. (2003). Pitch adaptive windows for improved excitation coding in low-rate CELP coders. IEEE Transactions on Speech and Audio Processing 11(6): 648-659*
- *Takehiro Sugimoto, & Shuichi Aoki, &Tomomi Hasegawa, & Tomoyasu Komori (2019). Advancement of 22.2 Multichannel Sound Broadcasting Based on MPEG-H 3-D Audio. IEEE Transactions on Broadcasting, PP (99):1*
- *Uttam Kr. Mondal (2019). Achieving lossless compression of audio by encoding its constituted components LCAEC. Springer and NASA Journal on Innovations in Systems and Software Engineering volume 15:75–85*
- *Wah Benjamin (2005). LSP-based multiple-description coding for real-time low bit-rate voice over IP. IEEE Transactions on Multimedia 7(1):167-178*
- *Wu Chou, & Bing Huang Juang (2002). Pattern Recognition in Speech and Language Processing.FL, United States: CRC Press, Inc.*
- *Yanzhen Ren; Hanyi Yang; Hongxia Wu; Weiping Tu; Lina Wang (2019). A Secure AMR Fixed Codebook Steganographic Scheme Based on Pulse Distribution Model. IEEE Transactions on Information Forensics and Security, 14(10):2649-2661*