A proposed fast ACELP codebook search

A. Falahati, M. Soleimani
Iran University of Science and Technology (IUST)
Faculty of Communication Engineering
afalahati@iust.ac.ir, mojtaba.soleimani@gmail.com

Topics: Coding and Modulations, Multimedia Communications

Abstract

The ACELP coding algorithm has been widely adopted in standard speech coders due to the advantages of codebook storage and the efficient search schemes. Especially, the focused search and depth first tree search that are very efficient methods to reduce dramatically the search complexity while providing a good sub optimal solution. The pulse location prediction and pulse replacement are already proposed to reduce complexity and provide improvements. In this paper, we propose two fast methods based on pulse replacement and pulse location prediction. In these proposed methods, the total computations are reduced with almost imperceptible degradation in performance and even better efficiency for large number of code replacements. The efficiency and complexity of proposed methods are examined on GSM-EFR coder and compared with depth first tree, pulse replacement and pulse location prediction methods.

Keywords: GSm-EFR, ACELP, Pulse replacement and Pulse location prediction

1. Introduction

For efficient speech communication, the code-excited linear prediction (CELP) structure is widely adopted by low-bit-rate speech coders [1]. Conventional CELP guarantees good quality at low bit rate by using the analysis by synthesis method. This method requires a large amount of memory for the code vectors and a great number of computations for the codebook search. In recent year, the Algebraic CELP (ACELP) algorithm developed to cope these problems has been widely adopted by standard speech coders such as ITU G.729 CS-ACELP [2, 3] and G.723.1 [4] for VoIP, AMR [5] for IMT-2000 systems, and IS-127 EVRC [6] for CDMA cellular systems are adopted as the standard speech codecs. The ACELP coding structure is popular due to the embedded efficient search for the optimal solution and no actual storage of the codebooks.

Generally, the ACELP structure needs to find the best combination of pulses and their corresponding signs from several fixed tracks to characterize the optimal speech excitation for minimizing the weighted mean square error. However, the ACELP search procedure still requires plenty of computational loads with the full search method to obtain a globally optimized excitation vector. Various fast methods are proposed to reduce computational complexity by decrease of pulse combinations. The focused search [7, 8] and depth first tree search [2, 3] algorithms are known as very efficient search approaches to reduce the search computational complexity.

In [9], a method based on ‘’pulse replacement’’ is introduced. At first, a number of coarse initial codevectors are searched sequentially with minimal search load. Then, the contribution of each pulse in codevector is measured and the least important pulse is removed from the codevector and a new pulse is searched and added. This procedure may be repeated to further enhance the performance of codevector. This method has the performance equivalent with depth first tree search with less codebook search load. The [10] proposes a fast ACELP algorithm using a designed pilot function to predict predetermined candidate pulses. With candidate pulses, this method not only reduces the number of search loops but also avoids the computation of unnecessary correlation functions.

In this paper we propose two combinational methods originated from pulse location prediction and pulse replacement methods by an additional adaptive threshold. These proposed methods reduce the computational load of pulse replacement method by elimination of computational load of selection of initial codevectors. One of proposed method decreases the total computations in replacing stage, also. In compare with pulse replacement, these proposed methods reduce the total computational complexity with almost imperceptible degradation in performance.

The efficiency and complexity of these two methods are compared with depth first tree, pulse-location prediction and pulse replacement methods in the example using the GSM-EFR coders. Section 2 describes the ACELP codebook algorithm, pulse-location prediction method, and pulse replacement method, while section 3 presents the two proposed methods based on combination of pulse location prediction and pulse replacement methods with an additional threshold for selection of replacing pulses. In section 4 by an example we present simulation results of two proposed methods and compare them with depth first tree search, pulse-location prediction, and pulse replacement methods. Finally, the conclusions are given in section 5.

2. ACELP codebook search

In ACELP coders, the algebraic codebook is searched by minimizing mean squared error between the weighted target speech and reconstructed speech as,

$$\varepsilon_k = \left\| x - gH\mathbf{c}_k \right\|^2 \qquad (1)$$

Where x is the target signal produced by subtracting the adaptive codebook contribution, g is the codebook gain, $H = h'h$,where h[n] is the impulse response of the vocal tract model, is defined as a lower triangular toeplitz convolution matrix with diagonal h(0) and the lower diagonals of h[l], h[2], . . . , and *h[L* – 1], where *L* is the size of subframe, and c_k is the algebraic code vector noted with the index k. it can be easily shown that minimizing (1) is equivalent to maximizing the term,

$$A_k = \frac{(C_k)^2}{E_{Dk}} = \frac{(H'x\mathbf{c}_k^t)}{\mathbf{c}_k^t H'H\mathbf{c}_k^t} = \frac{(\mathbf{d}^t\mathbf{c}_k)^2}{\mathbf{c}_k^t \Phi \mathbf{c}_k} \qquad (2)$$

Where $d = H^T x$ is known as backward filtered target representing the correlation function between the target signal *x[n]* and the impulse response *h[n]*, and $\Phi = H^T H$ is the correlation matrix of the impulse response. Since the algebraic codevector contains few nonzero pulses, the correlation in the numerator of (2) is given by,

$$C = \sum_{i=0}^{N_p - 1} v_i d(m_i) \qquad (3)$$

Where N_p is the number of pulses, m_i is the position of the i-th pulse and v_i is its signs. The energy in the denominator of (2) is

$$E_D = \sum_{i=0}^{N_p-1}\phi(m_i,m_i) + 2\sum_{i=0}^{N_p-2}\sum_{j=i+1}^{N_p-1} v_i v_j \phi(m_i,m_j) \quad (4)$$

Before the codebook search, the signal *d(n)* and the correlation $\Phi(i,j)$ are computed for (4) to reduce the computational complexity.

The depth-first tree search is used in the G.729A and the GSM-EFR coders to isolate the possible solution from a small set of most likely positions. In this approach, the positions of excitation pulses are partitioned into N subsets in a subframe. The depth first tree method, starts from global maximum point of the backward filtered signal and searches the trees along the path which has more chances to be the winner. For example, in the GSM-EFR, the 40 positions in each subframe are divided into 5 tracks where each track contains two pulses as shown in table 1. As a result, each excitation vector for each subframe contains 10 nonzero pulses with amplitudes +1 or -1.

Table 1. GSM-EFR algebraic codebook structure

Track	**Pulses**	**Positions**							
		R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7
1	i_0, i_5	0	5	10	15	20	25	30	35
2	i_1, i_6	1	6	11	16	21	26	31	36
3	i_2, i_7	2	7	12	17	22	27	32	37
4	i_3, i_8	3	8	13	18	23	28	33	38
5	i_4, i_9	4	9	14	19	24	29	34	39

In [9] a method based on pulse replacement is suggested. This method is a two-stage search structure. At the first stage, a number of coarse initial codevectors are sequentially searched by A_k values of (2) with minimal search load. Then at the second stage, pulse replacement procedure is applied to each initial codevector repeatedly to enhance the performance of codevector. In pulse replacement procedure, the measure of the contribution of each pulse is computed and the least important pulse is replaced with a new one.

The [10] proposes a method based on prediction of location of excited pulses. Prediction in this method is carried out by and designed pilot function. This pulse-location prediction reduces the number of candidate positions. Therefore, this method reduces the number of search loops and avoids the computation of inessential correlation functions.

3. Proposed fast codebook search method

The [9] proposes pulse replacement method for fast ACELP search. In this method one or more coarse initial codevector are selected by a sequential method. In next stage, pulse replacing is done for each codevector, separately. In this method, only one pulse is replaced in each replacing stage. In this paper, we propose a fast method in which the computational load of selection of initial codevectors reduces and the number of replacing pulse in each replacing stage is variable. The decrease of initial computational load is done by use of pulse location prediction, and the number of pulses that must be replaced in each stage is dynamically determined by an adaptive threshold. Finally, the computations of pulse replacing stage are reduced by use of primary pulse location prediction.

First goal of our proposed method is the decrease of computations of selection of initial codevectors. The overall computations of replacing method in [9], are $C\times\{L+(P+T)\times N\}$, where C is the number of initial codevectors, L is the search load for initial codevector, P is the number of pulses in codevector, T is the number of pulse positions in valid track, and N is the number of pulse replacements. Hence, the 'L' is an effective parameter in overall computations of pulse replacement method. By use of pulse location prediction, the number of candidate position of excited pulses is reduced. In pulse location prediction method the resolution of target signal (*r(n)*) is decreased. The lower resolution is obtained by a condensing pilot function. If we partition a subframe with L samples into K regions, the pilot function to characterize the energy of the target signal in each region is defined as

$$E(k) = p_k \sum_{n\in R_k} |r(n)|,\ k = 0,1,2,\ldots,K-1 \qquad (5)$$

Where $P_k = \left(\sum_{n\in R_k} r(n)\right)$ represents the polarity of *E(k)* in the *k*th region, and R_k is the partition region shown in table 1. Finally, the larger the amplitude of *E(k)* is, the higher possibility of the searched pulse in the *K*th region will be. Hence, we actually do not need to consider the sign of the pilot function since the sign of the search pulse can be determined by $sign(d(i))$. After calculation of pilot function we can select M region with larger absolute magnitude of *E(k)*, so the pulse positions are reduced to *ML/K* candidate locations. The selection of initial codevector is only done for these M regions. The M can be selected constantly or dynamically by a designed adaptive threshold.

After selection of initial codevectors, pulse replacing stage must be carried out for each initial codevector, separately. The contribution measure of each pulse in initial codevector is measured by A_k value of (2) after removing the corresponding pulse. By considering such computations, the higher A_k value of remaining codevector for each pulse removing indicate lower contribution measure of correspondin pulse and vise versa. Now, the least important pulses must be removed for replacing. We assume that each initial codevector has P pulses. The A_k values of each remaining codevector are arranged in descending order, with $A_k(\mathrm{P})$ as the largest and $A_k(1)$ as the smallest. The adaptive threshold for determination of the number of pulses replacing in each step is define as

$$\theta = \frac{1}{n}\sum_{i=0}^{n-1} A_k(p-i) \qquad (6)$$

This threshold is the average of n number of least important pulses that can be selected arbitrarily. The replacement must be done for each pulse that has higher A_k value than θ. The number of pulses that must be replaced is shown by β. The pulse replacing carries out one by one for each replacing pulse in ascending order of importance, so it begins from replacing of least important pulse (that has the largest A_k value). For example, if n=3 in (6) then β can be equal to one or two. If $\beta = 1$, after removing the least important pulse, a new pulse is searched from the valid track so that the A_k of new codevector is maximized. If $\beta = 2$, the replacement is firstly done for the least important pulse and then for another one. The replacing procedure for the second pulse also is similar to that of the first, but by considering that the first pulse is replaced by new optimum pulse from valid track.

Since the value of A_k never decreases by pulse replacement, the codevector approaches to the optimal solution steadily. If A_k does not change, which is the case when the removed pulses are selected again, the pulse replacement procedure is terminated. The block diagram of our proposed search method is depicted in Fig.1.

We apply the GSM-EFR coder for an example. As seen in table 1, there are 10 pulses in each codevector and 8 number of regions for condensing pilot function from R_0 to R_7, and each subframe has L=40 sample. The detailed procedure of our method is as follows.

- If the sum of normalized backward filtered target signal and normalized long-term prediction residual signal is *b(n)*, the pulse positions with maximum absolute values of *b(n)* are selected for each of the five track. Among the 5 maximums obtained for each track, the track with the highest value between five maximums is considered as global maximum, and 4 other maximums are known as local maximums.
- The pilot function of (5) is computed for the target signal containing 40 samples. The target signal converts to the lower resolution including 8 regions.
- The M regions with higher amplitude of E(k) are selected among 8 pulses. The M can be selected constantly or dynamically by an adaptive threshold. Since each region is contained of 5 pulses and we must totally select P=10 pulses for each codevector, the value of M must be equal or larger than 2 ($M \geq 2$).we select M=5, here.

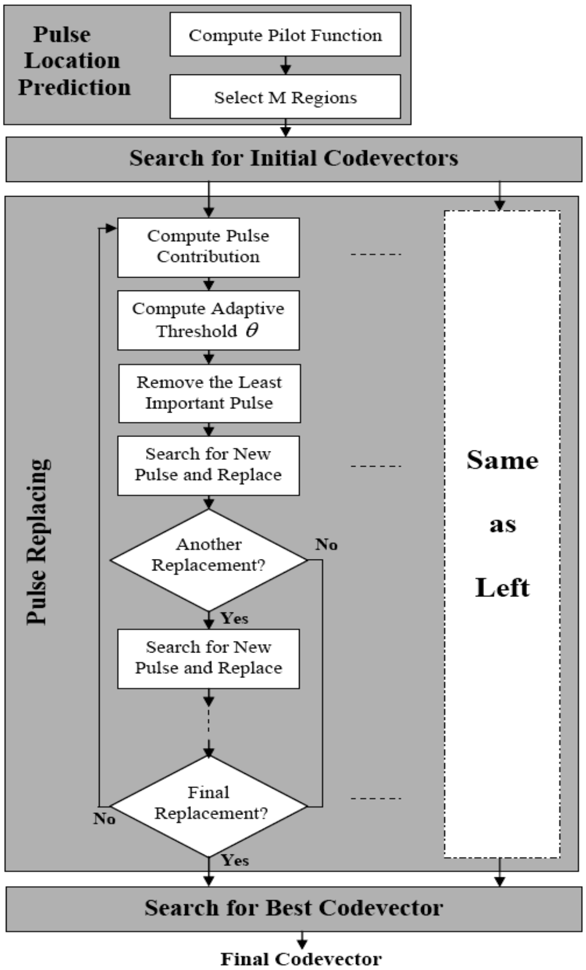

Fig. 1. Block diagram of proposed search method

- By considering the five obtained maximums from b(n), the first pulse i0 is always set in the position corresponding to the global maximum value and the position of i1 pulse is set in the maximum position of the next track. So, 4 initial codevectors is considered every which has a constant position of i0, and i1 is one of 4 local maximum for each of them.
- The other 8 pulses are arranged by considering the M selected regions in pulse location prediction stage. It means that the sequential search for selection of pair pulses must be done for the M selected regions pulses. So, the remaining 8 pulse positions are searched by two-track-based sequential search only for M regions. The computational complexity for each initial codevector in this stage is equal to $(M \times M) \times 4$.
- The pulse replacing process is carried out in this stage for 8 remaining pulses (i2 to i9) by considering constant position for i0 and i1 pulses. The contribution of all 8 pulses is separately calculated by removing each of them and computing A_k value of remaining 9-pulses codevector. Reasonably, the larger A_k value obtained from the remove of a pulse means to less contribution of it. We arrange the 8 obtained A_k values so that the largest and smallest be equal to $A_k(8)$ and $A_k(1)$, respectively. Now, the adaptive threshold is calculated by (6) (P=8). The n in (6) is select by designer, and we select n=3 for example. The number of codevectors with larger A_k than threshold (β) is considered as least important pulses and must be replaced with corresponding valid track pulses.
- The least important pulse is removed, at first. It replaced with each of 8 valid track pulses one by one, and A_k value of 8 different 10-pulse obtained codevector is computed. Corresponding pulse of Each of these obtained codevector that has largest value is replaced as optimum pulse with removed pulse.
- If $\beta = 2$, the replacing process is done for second pulse by considering that the first pulse is replaced by new optimum valid track pulse
- After replacing of least important pulses in stage one, searching for least important pulses and replacing of pulses larger than threshold is repeated. Of course if A_k does not change for each pulses, which is the case when the removed pulse are selected again, the pulse replacement procedure is terminated.
- After parallel processing for each of 4 initial pulses, 4 optimum output codevectors are obtained. Among these, the codevector that has the largest A_k is selected as final output.

The total computations of our proposed method can be further reduced by use of *M* regions of pulse prediction in pulse replacing stage, i.e. after computing the contribution of each pulse in codevector and removing the least important pulses, the replacement of each pulse is done from valid track but only for *M* pulses of it. For example, in GSM-EFR that each track has 8 pulses, only *M* pulses are searched for obtaining the best valid track pulse. Of course, the *M* regions contributing in selection of initial codevectors and the *M* regions of replacing stage can be selected differently. As a result, the computations of replacing stage are reduced. We name this second method that the *M* regions are used in replacing stage as 'reduced computations method' and first method as 'fast method'.

4. Computational complexity and simulation results

Both of our proposed methods reduce the total complexity of pulse replacement method. In fast method, the primary complexity of pulse replacement method is reduced by pulse location prediction. In this method, the number of replaced pulses (β) in each replacing stage varies based on the value of threshold. Therefore, the total computations for each subframe are changeable. For computing the total computations, this method is carried out for F subframes and the average of total computations is calculated. The average of total computations for each subframe is obtained as follow

$$\overline{X} = C \times \left\{ \left(M \times M + P \times N \right) + T \times \sum_{t=1}^{N} \overline{\beta}_t \right\} \qquad (7)$$

Where *M* is the number of selective regions that we consider constant value for it, *C* is the number of selective initial codevectors, *P* is the number of existing pulses of each codevector, *T* is the number of existing pulses in each track, *N* is the number of pulse replacing, and $\overline{\beta}_t$ is the average of β in each stage of pulse replacing (*N*).

If we use the pulse replacement method for GSM-EFR, the search load for initial codevector is $L = 8\times 8$, whereas this initial load for our fast method is equal to $L = M\times M$. It is obvious that the initial load for selection of each initial codevector is reduced based on the value of M.

In reduced computations method, in addition to reduction of the initial search load, the complexity of replacing stage is reduced, too. The total computations of this method is equal to

$$\overline{X} = C\times\left\{\left(M\times M + P\times N\right) + M\times\sum_{t=1}^{N}\overline{\beta}_t\right\} \qquad (8)$$

Therefore, the total computations of this method are further reduced in compare with the proposed fast method.

Both of our proposed methods are applied on the GSM-EFR coder and for *F*=10000 subframes of speech signal. For all tables and figures, we select 4 initial codevectors, n=3 in (6), and constant value for *M* for both proposed methods. Although $\overline{\beta}_1, \overline{\beta}_2, \ldots, \overline{\beta}_N$ have different values, the average of these values in each stage almost is equal to 1.5. In compare with pulse replacement method, because of the higher number of replacing pulses in replacing stage for our proposed methods (for pulse replacement only one pulse is replaced in each stage), it is reasonable that the total computations for our proposed methods increases with higher rate. Table 2, makes a comparison between the pulse replacement method and our two proposed methods on the basis of complexity, for some constant values of *M* and different *N*s. It is obvious in this table that for low number of replacement (*N*), the total computations is considerably reduced based on *M* selected regions.

Table2. Comparison between the complexity of the pulse replacement method and our two proposed methods (by considering n=3)

		N=0	N=1	N=2	N=3	N=4
Pulse replacement method		256	320	384	448	512
Fast method	**M=4**	64	148.3	224.2	303.9	382.4
	M=5	100	182.4	259.2	336.9	421.8
	M=6	144	224.8	302.6	386.2	462.5
	M=7	196	278.4	354.9	436.7	510.8
	M=8	256	337.5	416.1	494.7	574.3
Reduced computations method	**M=4**	64	121.4	178.2	230.9	288.2
	M=5	100	160.1	225.7	288.6	346.3
	M=6	144	214.3	277.9	349.5	416
	M=7	196	272.3	342.8	418.2	491.7
	M=8	256	337.5	416.1	494.7	574.3

In table 3, the methods of table 2 are compared on the basis of efficiency by SNR and segmental SNR for *M*=4, 6, 8. It can perceive from table 2 and 3 that the total computations are significantly reduced for low value of *M* and *N*, but the efficiency decreases, also. With increasing the values of *M* and *N*, the efficiency moves to higher value by some increasing of computational load.

Table3. Comparison between the efficiency of the pulse replacement method and our two proposed methods (by considering n=3)

			N=0	N=1	N=2	N=3	N=4
Pulse replacement method		**SNR**	12.43	13.38	14.02	14.31	14.39
		Seg SNR	10.94	12.46	13.48	13.93	14.01
Fast method	**M= 4**	**SNR**	11.53	13.02	13.84	14.26	14.44
		Seg SNR	10.08	12.02	13.31	13.88	14.08
	M= 6	**SNR**	12.15	13.22	13.98	14.34	14.49
		Seg SNR	10.66	12.35	13.43	13.98	14.13
	M= 8	**SNR**	12.43	13.54	14.17	14.41	14.51
		Seg SNR	10.94	12.64	13.61	14.04	14.12
Reduced computations method	**M= 4**	**SNR**	11.53	12.85	13.62	13.95	14.05
		Seg SNR	10.08	11.94	13.04	13.54	13.71
	M= 6	**SNR**	12.15	13.18	13.93	14.23	14.34
		Seg SNR	10.66	12.32	13.39	13.89	13.98
	M= 8	**SNR**	12.43	13.54	14.17	14.41	14.51
		Seg SNR	10.94	12.64	13.61	14.04	14.12

The tables 4 and 5 make a comparison between our two proposed methods, pulse location prediction method and depth first tree search on the basis of total complexity and efficiency, respectively. By considering the complexity and efficiency for different *N*, it seems that *N*=3 is an appropriate value, and we consider this value for table 4 and 5. The total computations of pulse location prediction method applied on GSM-EFR coder is equal to $4\times 4\times\left(M\times M\right)$ for different M selected regions. Also, we know that the total computations of depth first tree search method in GSM-EFR are equal to $4\times 4\times\left(8\times 8\right) = 1024$. According to these tables, not only the complexity of our proposed methods is lower than pulse location prediction and depth first tree search but also their efficiency is higher.

Table4. Comparison between the complexity of our two proposed methods with other mentioned methods, by considering n=3 and N=3

	M=4	M=5	M=6	M=7	M=8
Pulse location prediction	256	400	576	784	1024
Fast method	303.9	336.9	386.2	436.7	494.7
Reduced computations method	230.9	288.6	349.5	418.2	494.7
Depth first tree search	1024				

Table5. Comparison between the efficiency of our two proposed methods with other mentioned methods, by considering n=3 and N=3

		M=4	M=5	M=6	M=7	M=8
Pulse location prediction	**SNR**	13.21	13.71	13.99	14.13	14.22
	Seg SNR	12.79	13.35	13.68	13.81	13.89
Fast method	**SNR**	14.26	14.31	14.34	14.37	14.41
	Seg SNR	13.88	13.94	13.98	14.01	14.04
Reduced computations method	**SNR**	13.95	14.12	14.23	14.34	14.41
	Seg SNR	13.54	13.74	13.89	13.98	14.04
Depth first tree search	**SNR**	14.22				
	Seg SNR	13.89				

Fig.2 depicts the total computations of our proposed methods and other mentioned method for some optimum values of *M* selected regions by considering *N*=3.

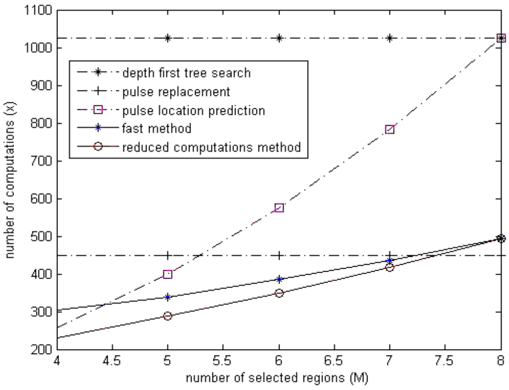

Fig. 2. Comparison between complexity of our proposed methods (solid lines) and other methods (doted line), by considering N=3 and n=3

Fig.3 makes a comparison between the efficiency of our proposed methods and other mentioned method on the basis of SNR and for some values of *M*, by considering *N*=3.

In Fig.4 the complexity of several methods is compared for all mentioned methods for different value of *N*, by considering *M*=6 that is a good selection (according to efficiency and complexity) for our proposed methods.

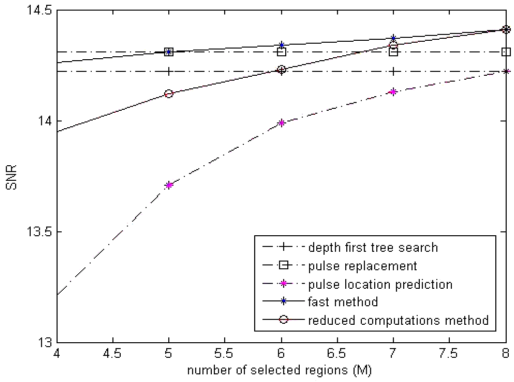

Fig. 3. Comparison between efficiency of our proposed methods (solid lines) and other methods (doted line), by considering N=3 and n=3

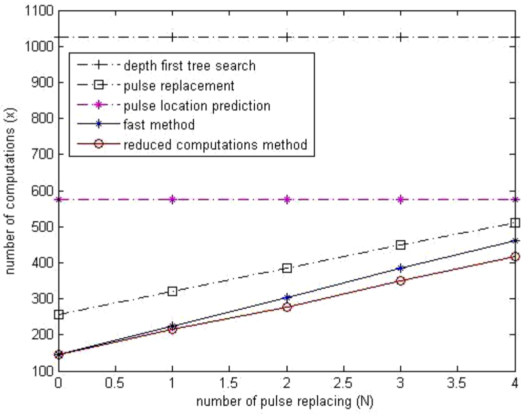

Fig.4. Comparison between complexity of our proposed methods (solid lines) and other methods (doted line), by considering M=6 and n=3

In Fig.5 the efficiency of all methods is compared for various *N*s, by considering *M*=6.

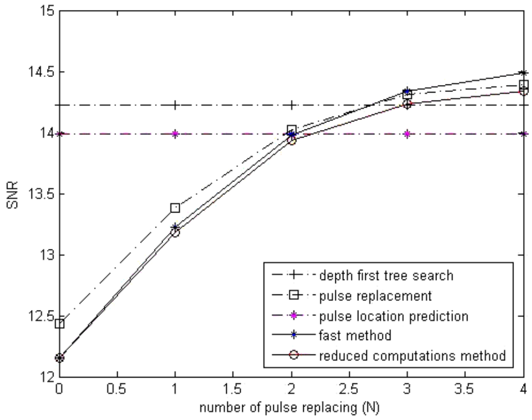

Fig.5. Comparison between efficiency of our proposed methods (solid lines) and other methods (doted line), by considering M=6 and n=3

According to Fig. 2-5, it can be said that the selection of *N*=3 and *M*=6 are reasonable in the sense of complexity and efficiency. As seen in Fig.2 and 3, in compare with pulse location prediction method, our proposed methods not only have lower complexity but also have a better efficiency. By considering Fig.4 and 5, it is seen that the efficiency of our proposed methods is near the pulse replacement method but by lower complexity. For $N \geq 3$, the fast proposed method has the lower complexity and higher efficiency than pulse replacement method. In addition to mentioned points of reduced computations method, some of unnecessary correlation functions are not computed, also.

5. Conclusion

Two methods containing fast and reduced combinations methods for improving the ACELP fixed codebook search are proposed. The usefulness of proposed fast search methods is examined by implementing the GSM-EFR codec. In comparison with the pulse replacement method, these proposed method, by having almost equal efficiency, have lower complexity and with increasing the number of pulse replacing can even have higher efficiency than pulse replacement method. With respect to pulse location prediction, these proposed methods not only have lower complexity but also have better efficiency. Both of our proposed methods are based on pulse replacement and pulse location prediction methods. In the fast method, by using pulse location prediction and selection of some of all regions, the complexity of selection of initial codevectors is reduced. In replacing stage, the number of replacing pulses in each step is variable and select by using an adaptive threshold. In reduced computations method, in addition to having the changes of fast method, the replacing is only done for the primary number of selected regions chosen by pulse location prediction. So, the overall complexity is further reduced in this method. For low number of pulse replacing and primary selected regions, the efficiency is reduced with respect to pulse location prediction and pulse replacement methods, but it can be compensated by increasing the number of pulse replacements. These proposed methods are compatible with any ACELP type codec.

Acknowledgments

The authors thank the Iran Telecommunication research center (ITRC) for its scientific and financial support.

References

[1] A. M. Kondoz, “Digital Speech”，Wiley Publisher, pp. 135, 1994.

[2] ITU-T Recommendation (G.729-Annex E: ‘11.8 kbit/s CS-ACELP speech coding algorithm’, September 1998

[3] SALAMI, R., LAFLAMME, C., BESSETTE, B., and ADOUL, J.-P.: ‘ITU-T ‘G.729 Annex A: Reduced complexity 8 kb/s CS-ACELP codec for digital simultaneous voice and data’, IEEE Commun. Mag., 35, 1997

[4] ITU-T Recommendation (G.723.1: ‘Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s’, March 1996

[5] ETSI EN 301 703, “Digital cellular telecommunication system (Phase 2+); Adaptive Multi-Rate (AMR)”, 1999.

[6] TIAIEIMS-lZ7, “Enhanced variable rate codec; speech service option 3 for wideband spread-spectrum digital systems,” 1997.

[7] ETSI: ‘Digital cellular telecommunications system (Phase 2); Enhanced full rate (EFR) speech transcoding (GSM 06.60 version 6.0.0)’, ETSI EN 300 726,1997

[8] SALAMI, R., LAFLAMME, C., BESSETTE, B., and ADOUL, J.-P.: ‘Description of GSM enhanced full rate speech codec’. Proceedings of IEEE international conference on Communications, ICC ’97, Montreal, Canada, 1997, vol. 2, pp. 725-729

[9] Hochong Park, Younchang Choi and Doyoon Lee, “Efficient Codebook Search Method for ACELP Speech Codecs”, IEEE , 2002.

[10] F.K.Chen, J.F.Yang and Y.L.Yan, “Candidate Scheme for Fast ACELP Search”, IEE Proceeding on Image Signal Processing , February 2002.