SPARSE SIGNAL APPROXIMATION ALGORITHMS IN A CELP CODER

Przemysław Dymarski and Rafał Romaniuk

Institute of Telecommunications, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665, Warsaw, Poland
email: dymarski@tele.pw.edu.pl, r.romaniuk@tele.pw.edu.pl

ABSTRACT

This work presents a series of sparse signal modeling algorithms implemented in a typical CELP coder in order to compare their performances at a reasonable computational load. New algorithms are proposed, based on cyclic and parallel use of a fast implementation of the optimized orthogonal matching pursuit algorithm, i.e. the recursive modified Gram – Schmidt algorithm. These algorithms yield a statistically significant reduction of signal approximation error at a controllable computational complexity.

1. INTRODUCTION

The idea of CELP coder emerged in the early 80's [1], when the possibilities of real time implementation were limited. The analysis-by-synthesis algorithms had to be simplified to enable successful implementations and setup of international standards. Most of these standards (e.g. [2]) appeared in the 90's but nowadays the technical progress in microprocessors and programmable devices enables implementation of much more complex speech coding algorithms. On the other hand, in the last decade rapid development of sparse approximation and compressive sensing techniques is observed, yielding the complementary matching pursuit [9], [20], cyclic matching pursuit [11], least angle regression [10], basis pursuit [14], subspace pursuit and many other algorithms (ref. [8] to[18]).

The aim of this paper is to take up the challenge of comparing some of the sparse approximation algorithms and to propose new ones which may be implemented in a typical CELP coder. The CELP coder used for testing is the G.723.1 coder [2], [21] with the algebraic code excited linear prediction algorithm replaced by a series of sparse signal modelling algorithms.

This paper is organized as follows: In Sect.2 the problem is posed and the notation is introduced. In Sect.3 the selected sparse approximation algorithms are overviewed and new algorithms are proposed, in particular the multi – recursive modified Gram - Schmidt algorithm. In Sect.4 the results of simulations are presented and in Sect.5 they are briefly summed up.

2. SIGNAL MODELING IN CELP CODERS

The CELP coder (Fig.1) may be regarded as a multistage vector quantizer with filtered codebooks.

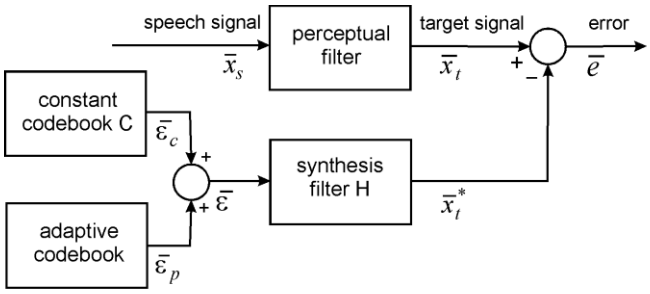

Fig.1. Generic scheme of the CELP coder

The N-dimensional vector of the perceptual speech signal $\bar{x}_t$ is modeled using the filtered vectors issued from two codebooks (the adaptive one and the constant one) so as to minimize the error $||\bar{e}||^2 = ||\bar{x}_t - \bar{x}_t^*||^2$. The linear predictive filter $H(z)$ is used for filtering of the signal issued from both codebooks, which may be described as $\bar{x}_t^* = \bar{x}_0 + H\bar{\varepsilon}$, where $\bar{x}_0$ is the zero input response, and H is the lower triangular Toeplitz matrix built on the impulse response of $H(z)$. The perceptual signal model may be described as $\bar{x}_t^* = \bar{x}_0 + \bar{x}_c + \bar{x}_p$, where $\bar{x}_c = H\bar{\varepsilon}_c$ is the signal issued from the constant codebook and $\bar{x}_p = H\bar{\varepsilon}_p$ is the signal issued from the adaptive codebook (long term prediction). Assuming that the long term prediction signal is known (in this paper we do not optimize this stage of signal modeling), the error energy may be rewritten as follows: $||\bar{e}||^2 = ||\bar{x}_t - \bar{x}_0 - \bar{x}_p - \bar{x}_c||^2 = ||\bar{x} - \bar{x}^*||^2$, where $\bar{x} = \bar{x}_t - \bar{x}_0 - \bar{x}_p$ - the target signal and $\bar{x}^* = \bar{x}_c$ - its model.

Having the codebook $C = [c^1, c^2, \ldots, c^L]$, consisting of L N-dimensional vectors, the excitation signal $\bar{\varepsilon}_c$ is usually obtained with the use of one of the two approaches. Both may be described with the formula $\bar{\varepsilon}_c = C\bar{g}$, but in the multipulse or classical CELP coders $\bar{g}$ is a L-dimensional vector of gains, $K<N$ of which are nonzero, and in ACELP coders the nonzero gains are quantized using 2-level symmetrical quantizer (i.e. only two gain values, $\pm g$, are allowed). In this paper only the first approach will be analyzed, the optimization of the ACELP scheme will be left for future research. By filtering of the excitation signal $\bar{\varepsilon}_c$, the target signal model $\bar{x}^* = \bar{x}_c$ is obtained: $\bar{x}^* = H\bar{\varepsilon}_c = HC\bar{g} = F\bar{g}$,

© EURASIP, 2011 - ISSN 2076-1465

where $F = HC = [f^1, f^2, \ldots, f^L]$ - the filtered codebook. The aim of the modeling procedure is to minimize the error while keeping the degree of sparsity equal to K, i.e. only K gains may have nonzero values:

$$\min\|\bar{x} - F\bar{g}\|_2^2, \qquad \|\bar{g}\|_0 = K \tag{1}$$

There are three tasks associated with the problem (1):

1. Construction of the codebook C – in the multipulse coders it is the unit matrix, in CELP coders it may be stochastic or calculated using the training speech data. In fact, any codebook containing normalized vectors uniformly distributed in N-dimensional space yields similar results.
2. Filtering – the classical predictors yielding the minimum energy of prediction error (min of L2 norm) are usually used for construction of the filter $H(z)$, but recently the sparse predictors were proposed [12,13], which enhance sparsity of the residual signal. Thus, better signal models are obtained using a small number of vectors (K).
3. Codebook search – this problem exists for any codebook C and any filter H and will be considered in this paper. Sparse approximation algorithms will be tested, but it must be noticed that the problem is not really sparse, i.e. $K=N$ vectors are necessary to obtain $||\bar{e}||^2 = 0$.

3. SPARSE APPROXIMATION ALGORITHMS

3.1 The matching pursuit and relative algorithms

The problem (1) is NP-hard, so its optimal solution requires exhaustive search of all K-combinations of L codebook vectors. The number of such combinations is $\binom{L}{K}$, which in typical applications is definitely prohibitive. Therefore, sub-optimal approaches are used, particularly the greedy algorithms, in which the codebook vectors are selected one by one and cannot be removed. In the **matching pursuit** (MP) algorithm successive projections of the error vector on the nearest (according to angle criterion) codebook vector are made. The MP algorithm has been discussed as one of possible solutions to the problem of the excitation signal generation in multipulse predictive coders ([5,6]), in [3] it has been applied to audio coding. Having the target signal $\bar{x}$ and the filtered codebook F, the squared norms $\alpha^j = ||f^j||^2$, and the scalar products (correlations) $\beta_1^j = <\bar{x}, f^j> = \bar{x}^T f^j$ are calculated. The first vector $f^{j(1)}$ is then selected, which maximizes $(\beta_1^j)^2/\alpha^j$, i.e. the squared norm of the orthogonal projection of $\bar{x}$ on f^j. The optimal gain associated with this vector equals $g_1 = \beta_1^{j(1)}/\alpha^{j(1)}$ and the error vector $\bar{e}_1 = \bar{x} - g_1 f^{j(1)}$ is orthogonal to $f^{j(1)}$. The correlations $\beta_2^j = <\bar{e}_1, f^j>$ are updated using the formulas $\beta_2^j = \beta_1^j - r_1^j \beta_1^{j(1)}/\sqrt{\alpha^{j(1)}}$, $r_1^j = <f^j, f^{j(1)}>/\sqrt{\alpha^{j(1)}}$. Thus, the second iteration is prepared, in which the vector $f^{j(2)}$ is chosen, maximizing $(\beta_2^j)^2/\alpha^j$, etc. In K steps a sequence of vectors and gains is obtained, but the error vector $\bar{e}_K$ is not orthogonal to the subspace spanned by the selected vectors. The optimal gains corresponding to the selected vectors may be recalculated using the formula $g = (A^T A)^{-1} A^T \bar{x}$, where $A = [f^{j(1)}, \ldots, f^{j(K)}]$ is a matrix consisting only of the selected vectors (if they are distinct).

If this gain updating is made at the end of any iteration of the matching pursuit algorithm, then we obtain the **orthogonal matching pursuit** algorithm (OMP), which has been widely used in the multipulse and CELP coders [5,6,7], then adapted to audio coding [4]. However, this approach does not exhibit the local optimality property, because vector $f^{j(k)}$, selected in the k-th step (according to the minimum angle criterion) is not necessarily the best vector, minimizing the projection error of the target vector on a subspace spanned by the vectors $\{f^{j(1)}, f^{j(2)}, \ldots, f^{j(k)}\}$.

In order to assure the locally optimal selection, the codebook should be orthogonalized with respect to the previously chosen k-1 vectors before selecting the k-th one. Thus, we obtain the **optimized orthogonal matching pursuit** algorithm (OOMP), suggested for the multipulse coders [6], then applied in a CELP coder [7], finally formalized in [8]. The successive codebook orthogonalization is a time consuming task, but there is a fast OOMP implementation, which has been proposed under the name of **Recursive Modified Gram-Schmidt** (RMGS) algorithm [7,19]. This implementation consists in a virtual codebook orthogonalization, i.e. processing does not apply to vectors, but only to their norms and correlations, which requires about KNL operations, as in the matching pursuit algorithm. This algorithm will be described briefly because it is a base of a newly proposed M-RMGS algorithm. The first step does not differ from the matching pursuit: squared norms of the original (not yet orthogonalized) codebook vectors $\alpha_1^j = ||f^j||^2$ and scalar products $\beta_1^j = <\bar{x}, f^j>$ are calculated and the first vector $f^{j(1)}$, which maximizes $(\beta_1^j)^2/\alpha_1^j$ is selected. In the next step, the whole codebook is virtually orthogonalized with respect to $f^{j(1)}$ (i.e. $f^j_{orth(2)} \perp f^{j(1)}$) by calculating the squared norms $\alpha_2^j = ||f^j_{orth(2)}||^2$ and correlations $\beta_2^j = <\bar{e}_1, f^j_{orth(2)}> = <\bar{x}, f^j_{orth(2)}>$ in a recursive way: $\alpha_2^j = \alpha_1^j - (r_1^j)^2$, $\beta_2^j = \beta_1^j - r_1^j \beta_1^{j(1)}/\sqrt{\alpha_1^{j(1)}}$, where $r_1^j = <f^j, f^{j(1)}>/\sqrt{\alpha_1^{j(1)}}$. Then the second (generally: the k-th) vector is selected, maximizing

$$\frac{\left\langle \bar{e}_{k-1}, f^j_{ortk(k)} \right\rangle^2}{||f^j_{orth(k)}||^2} = \frac{\left\langle \bar{x}, f^j_{ortk(k)} \right\rangle^2}{||f^j_{orth(k)}||^2} = \frac{\left(\beta_k^j\right)^2}{\alpha_k^j} \tag{2}$$

and the virtual orthogonalization with respect to this vector ($f^{j(k)}_{orth(k)}$) is performed:

$$\alpha_{k+1}^j = ||f^j_{orth(k+1)}||^2 = \alpha_k^j - (r_k^j)^2 \tag{3}$$

$$\beta_{k+1}^j = <\bar{x}, f^j_{orth(k+1)}> = \beta_k^j - r_k^j \beta_k^{j(k)}/\sqrt{\alpha_k^{j(k)}} \tag{4}$$

where $r_k^j = \left[< f^j, f^{j(k)} > - \sum_{i=1}^{k-1} r_i^{j(k)} r_i^j \right] / \sqrt{\alpha_k^{j(k)}}$ (5)

Note that the number of operations is only slightly increased as compared to the matching pursuit (the main difference is in eq. (5)) – there are still about *KNL* operations.

3.2 The complementary matching pursuit algorithms

The complementary approach consists in finding a sparse approximation of the minimum norm solution $\bar{g}_{\min}$ of the underdetermined system $\bar{x} = F\bar{g}$ [9]. This leads to the following problem:

$$\min \| \bar{g}_{\min} - \Phi \bar{g} \|_2^2, \qquad \| \bar{g} \|_0 = K \tag{6}$$

where $\Phi = F^T (FF^T)^{-1} F$ and $\bar{g}_{\min} = F^T (FF^T)^{-1} \bar{x}$. The sparse solution may be obtained using the MP, OMP or OOMP algorithms, thus we have the **complementary matching pursuit** (CMP), **orthogonal complementary matching pursuit** (OCMP) and **optimized orthogonal complementary matching pursuit** (OOCMP) algorithms. The matrix Φ is of $L \times L$ dimension, but the squared norms of its columns α^j, the correlations β_k^j and the inner products r_k^j may be obtained out of the columns of the $N \times L$ -dimensional matrix $C_F = (FF^T)^{-1} F$ [9,19].

3.3 The COSAMP and Subspace Pursuit (SP)

The COSAMP and SP are non-greedy algorithms which tend to improve a given set of *K* codebook vectors by appending new ones (augmentation phase) and rejecting the other ones (optimization phase) [15,16]. The similar augmentation phase is also present in the stagewise orthogonal matching pursuit (StOMP) [17] and in the regularized orthogonal matching pursuit (ROMP) [18], but StOMP and ROMP are greedy, i.e. rejection of the selected vectors is not possible.

The augmentation phase consists in calculation of the correlations $\bar{\beta} = F^T \bar{e}$, where $\bar{e}$ is the error vector (in the first iteration $\bar{e} = \bar{x}$). Then the codebook vectors (rows of F^T) yielding the largest absolute values of the correlations are selected (2*K* in COSAMP and *K* in SP) and merged with the *K* previous ones (in the first iteration there is no merging). In the optimization phase the optimal gains are calculated, which minimize the norm of the error vector ($g = (A^T A)^{-1} A^T \bar{x}$, where *A* is a matrix containing the set of codebook vectors obtained in the augmentation phase). Then *K* vectors are selected, yielding the maximum absolute values of gains and the error vector is updated.

3.4 The algorithms based on L1 minimization

In order to apply the convex optimization methods, the L0 norm in problem (1) is replaced by the L1 norm. Various numerical methods may be used to solve this kind of problems. Some of them are greedy, e.g. the Least Angle Regression [10], consisting in minimization of the angle between the target vector $\bar{x}$ and a weighted sum of codebook vectors. These greedy algorithms do not give much better results than the basic MP algorithm (see e.g. [19]), therefore, the non-greedy formulations are used. In this work the following problem is considered [14,12]:

$$\min \left\{ \lambda \| \bar{g} \|_1 + \| \bar{x} - F\bar{g} \|_2^2 \right\} \tag{7}$$

The Basis Pursuit algorithm [22] has been used to solve (7), but there were problems in controlling the degree of sparsity $\| \bar{g} \|_0$. The degree of sparsity depends on the value of the parameter λ, so many trials are necessary to obtain the desired number of nonzero gains. In this work pruning is used in order to obtain exactly *K* nonzero gains and 10 trials are made, using different values of λ.

3.5 The cyclic optimization

The cyclic optimization, like COSAMP and SP, consists in substituting new codebook vectors for the previously chosen ones, but the substitution is made in a one by one manner and the subspace dimension *K* is kept constant. The cyclic MP algorithm was proposed in [11] and in [19] the cyclic optimization was combined with the OOMP and OOCMP algorithms. The algorithms described in [19] consist of an augmentation phase and an optimization phase. In the augmentation phase the optimized orthogonal matching pursuit (OOMP or OOCMP, with the fast RMGS implementation) is applied. Thus the initial *K*-dimensional subspace is obtained. The cyclic optimization phase is always based on the OOMP (using the RMGS algorithm). In the main loop the codebook vectors $f^{j(i)}, i = 1,2,\ldots,K$ are, one by one, temporarily removed from the subspace. The codebook vector (not belonging to the reduced subspace) is searched, which, when appended to the *K*-1 remaining vectors, yields the best approximation of the target vector $\bar{x}$ (minimum norm of the projection error vector). Frequently, it is just the vector which has been removed and there is no substitution. The algorithm stops if there are no substitutions in *K* consecutive trials, but it may be stopped at any moment if the number of operations attains a predefined value. Note that this algorithm yields, in any case, no worse results than the algorithm used in the augmentation phase.

3.6 The Multi-RMGS algorithm

The idea of this algorithm is to compute, in a parallel way, not only one locally optimal sequence of codebook vectors, but *M* sequences, yielding small approximation error. The subspace dimension gradually increases from 1 to *K*, but the number of sequences is equal to *M*. If *M*=1, the M-RMGS algorithm is equivalent to the OOMP (with the fast RMGS implementation).

At the first step the squared norms of the codebook vectors $\alpha_1^j = \| f^j \|^2$ and the scalar products $\beta_1^j = < \bar{x}, f^j >$ are calculated. Then, codebook vectors are sorted according to the criterion $\| \bar{e}_1^j \|^2 = \| \bar{x} \|^2 - (\beta_1^j)^2 / \alpha_1^j$, which is the squared norm of the error vector. The first *M* vectors (indices $j(1,1), j(1,2), \ldots, j(1,M)$) start *M* sequences. For each sequence *m*=1,..,*M* the values $r_{1,m}^j = < f^j, f^{j(1,m)} > / \sqrt{\alpha_1^{j(1,m)}}$, $\alpha_{2,m}^j = \alpha_1^j - (r_{1,m}^j)^2$ and $\beta_{2,m}^j = \beta_1^j - r_{1,m}^j \beta_1^{j(1,m)} / \sqrt{\alpha_1^{j(1,m)}}$ are computed, as in the RMGS algorithm. The squared error norms $\varepsilon_{1,m} = \| \bar{e}_1^{j(1,m)} \|^2$ are also retained for each sequence.

At the second step (generally, at the k-th step) there are almost ML possible sequences (to any of M sequences any of L-k+1 vectors may be appended), but only M of them are retained. The criterion is, of course, the squared norm of projection error: $\varepsilon^{j}_{k,m} = \varepsilon_{k-1,m} - (\beta^{j}_{k,m})^2 / \alpha^{j}_{k,m}$. Thus, the best vectors $j(k,1), j(k,2), \ldots, j(k,M)$ are selected, the corresponding squared error norms are updated and the operations described with formulas (3),(4),(5) are performed for each sequence. The identical sequences (permutations of the same vector indices) are not allowed. They are easily recognized because they yield the same squared error value. Many sequences started at the first step do not survive, sometimes all sequences considered at the last step stem from the same vector chosen at the first step. At the last step the best sequence is selected.

The results of the M-RMGS algorithm are not worse, in any case, than the results of the OOMP (RMGS) algorithm. The cyclic OOMP has the same property, but the advantage of the M-RMGS consists in its lower and constant computational complexity (about $MKNL$ operations), while the number of iterations of the cyclic OOMP is variable.

4. TESTING

The sparse approximation algorithms are usually compared using the synthetic signals, e.g. realizations of an uncorrelated or an AR random process [19,20]. In this work the tested algorithms are incorporated in the CELP coder and the real audio signals are used.

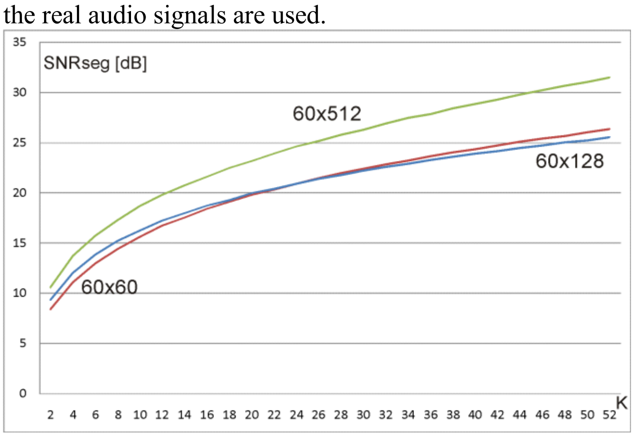

Fig.2 The segmental SNR [dB] for the MP algorithm

In the Matlab implementation of the G.723.1 coder [21] the ACELP modelling block has been removed and the described above algorithms of codebook vectors indices ($j(1),\ldots,j(K)$) and gains ($\bar{g}$) calculation are implemented. The long and short time predictors, perceptual filters, frame and subframe (N=60) lengths are left unchanged, except of postfilters, which could affect the SNR values. Three unfiltered codebooks C are used: a unit matrix of dimension 60x60 for the multipulse coder and two matrices of dimension 60x128 and 60x512 for the CELP coder. The CELP codebooks consist of the normalized vectors uniformly distributed on a 60-dimensional sphere. No gain quantization is applied in order to test only the vector selection algorithms.

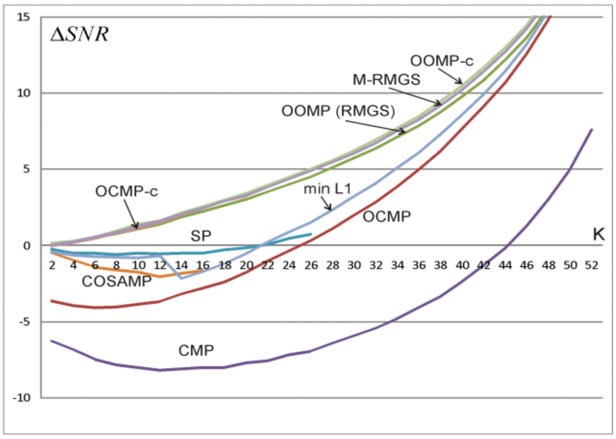

Fig.3. $\Delta SNR\,[dB]$ for the multipulse coder (codebook 60x60)

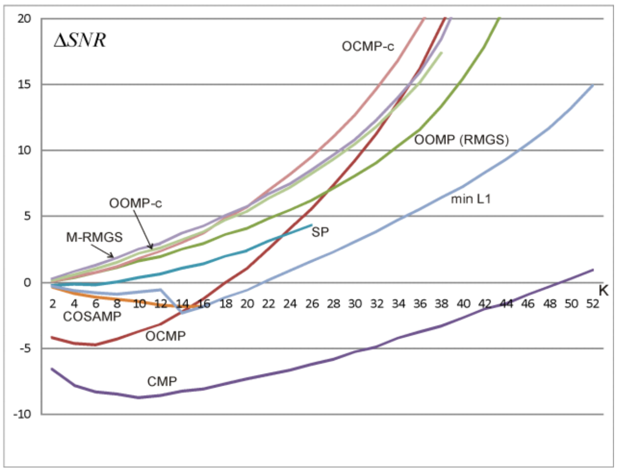

Fig.4. $\Delta SNR\,[dB]$ for the CELP coder with the codebook 60x128

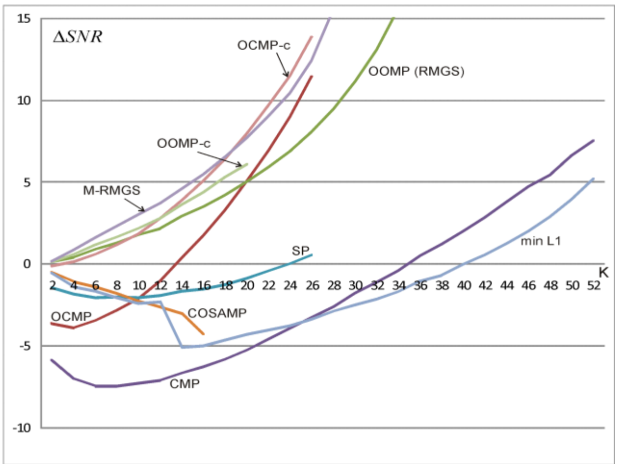

Fig.5. $\Delta SNR\,[dB]$ for the CELP coder with the codebook 60x512

Five audio files, containing male and female speech and a song are used for testing, thus we have obtained almost 2500 N=60-dimensional segments (subframes). For each segment the SNR_i [dB] is calculated and the mean value SNRseg is obtained. In Fig.2 the SNRseg values for the matching pursuit algorithm (MP) are shown for three above- mentioned codebooks and for different number (K) of codebook vectors used for signal approximation. These results are used as a reference for testing the other algorithms. In the following tests, the SNR_i values in each segment are compared with the corresponding values obtained for the MP algorithm, e.g. $\Delta SNR_i(\text{OOMP}) = SNR_i(\text{OOMP}) - SNR_i(\text{MP})\,[dB]$. This is because the absolute SNR_i values exhibit greater variance than the ΔSNR_i values. Then the mean values (e.g. $\Delta SNR(\text{OOMP})\,[dB]$) are calculated and the confidence interval is evaluated. The results are presented in Fig. 3,4,5. Note that the confidence interval (at the confidence level of 90%) is about 0.2 dB for small values of K and about 0.5 dB for K >30.

The results may be summarized as follows:

1. M-RMGS (here M=10) outperforms the other tested algorithms for small K (e.g. 10) and larger codebooks (L=512).
2. OCMP-c is the best at greater K (the results obtained for ARMA process [19] are thus confirmed). Its disadvantage is a higher computational load (many substitutions are necessary to obtain these good results).
3. Complementary algorithms (CMP, OCMP) perform very well if the problem is really sparse [9,19,20], because the matrix Φ (6) has nearly orthogonal columns, which facilitates finding the proper codebook vectors. For a non-sparse problem, they are interesting only for large K.
4. SP and COSAMP are designed to deal with sparse problems, so in our testing they do not perform very well.
5. As reported in [12,13], minimization of the L1 norm, when applied not only to gains (as in 7) but also to prediction coefficients, yields very promising results. In our tests we use the classical predictor and we have obtained satisfactory results only for greater K.
6. Complexity analysis (excluding the newly proposed M-RMGS) is given in [19]. For K=10 implementation of the MP, RMGS and M-RMGS (M=10) algorithm in the CELP coder requires 12.8, 14.7 and 159 Mflops for the 60x128 codebook and 51.2, 58.1 and 636 Mflops for the 60x512 codebook, correspondingly. Thus the commonly used DSP devices may be applied.

5. CONCLUSIONS

Development of microprocessors and programmable devices enables implementation of more complex speech coding algorithms. The proposed algorithms based on cyclic and parallel use of the fast optimized orthogonal matching pursuit algorithm (RMGS) offer a statistically significant improvement of the segmental SNR at a reasonable computational complexity. They may be used e.g. in the variable rate speech coders

REFERENCES

[1] B. Atal and J. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 614–617, 1982.

[2] *ITU-T Recommendation G.723.1*, Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s, March 1996.

[3] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[4] Y. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition" *Proc. 27th Asilomar Conf. Signals, Systems Comput.*, 1993

[5] M. Berouti, H. Garten, P. Kabal, and P. Mermelstein, "Efficient computation and encoding of the multi-pulse excitation for LPC," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 10.1.1–10.1.4, 1984.

[6] P. Kroon and E. Deprettere, "Experimental evaluation of different approaches to the multi-pulse coder," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 10.4.1–10.4.4, 1984.

[7] P. Dymarski, N. Moreau, and A. Vigier, "Optimal and suboptimal algorithms for selecting the excitation in linear predictive coders," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 485–488, 1990.

[8] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9, pp. 137–140, 2002.

[9] G. Rath and C. Guillemot, "Sparse approximation with an orthogonal complementary matching pursuit algorithm" *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 3325–3328, 2009.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," The Annals of Statistics, vol. 32, no. 2, pp. 407–499, 2004.

[11] M. Christensen and S. Jensen, "The cyclic matching pursuit and its application to audio modeling and coding," *Rec. Asilomar Conf. Signals Systems and Computers*, 2007.

[12] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Retrieving sparse patterns using a compressed sensing framework: Applications to speech coding based on sparse linear prediction," *IEEE Signal Processing Letters*, vol. 17(1), pp.103-106, 2010.

[13] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Moonen, "Enhancing sparsity in linear prediction of speech by iteratively reweighted 1-norm minimization", *Proc. ICASSP 2010*, pp. 4650-4653

[14] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001

[15] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. vol.26(3), pp.301-321, 2008.

[16] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol.55(5), pp.2230-2249, 2009.

[17] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," *Tech. Report, Department of Statistics, Stanford University*, vol. 2, 2006.

[18] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4(2), pp.310-316, 2010.

[19] P. Dymarski, N. Moreau and G. Richard, „Greedy sparse decompositions: a comparative study", submitted to *Eurasip Journal on Advances in Signal Processing*, 2011

[20] G. Rath and A. Sahoo, "A comparative study of some greedy pursuit algorithms for sparse approximation," *Eusipco,* 2009.

[21] P. Kabal "ITU-T G.723.1 Speech Coder: A Matlab Implementation", *McGill Univ.*, 2004

[22] G. Peyre: Toolbox sparsity - sparsity-based signal processing related functions, *Matlab Central*, 2007.