A Novel Approach for Supporting Approximation Representation of Linear Prediction Residuals by means of Simulated-Annealing-Based Optimization

Alfredo Cuzzocrea[1], Enzo Mumolo[2], and Giorgio Mario Grasso[3]

[1] DIA Dept., University of Trieste and ICAR-CNR, Italy
alfredo.cuzzocrea@dia.units.it
[2] DIA Dept., University of Trieste, Italy
mumolo@units.it
[3] CSECS Dept., University of Messina, Italy
gmgrasso@unime.it

Abstract. The goal of this work is to describe an optimization approach for selecting a reduced number of samples of the linear prediction residual. Sample determination is a combinatorial problem. Our approach addresses the combinatorial problem with simulated annealing based optimization. We show that better results than that obtained by a standard approximation approach, namely the multipulse algorithm, are obtained with our approach. Multipulse selects pulse locations by a sequential, sub-optimal, algorithm and computes the pulses amplitudes according to an optimization criteria. Our approach finds the optimal residual samples by means of an optimization algorithm approach without amplitudes optimization. The compressed residual is fed to an all-pole model of speech obtaining better results than standard Multipulse modeling. We believe that this algorithm could be used as an alternative to other algorithms for medium-rate coding of speech in low complexity embedded devices. In this paper we also discuss performance and complexity issues of the described algorithm.

1 Introduction

Research on approximation of the linear prediction residue has been an active field for many years in the past. Important results have been obtained with the multipulse approach [1, 2]. The multipulse approach has been extensively used to model the prediction residual in medium-rate speech coders. High-quality speech coders at bit-rates in the vicinity of 8 Kbits have been developed with the multipulse approach. However, it is very difficult to reduce the bit-rate with this approach, basically because of the quantization of the pulses amplitudes. CELP coders [3], which model the excitation signal with a binary sequence, can produce good-quality speech at bit rates as low as 4.8 Kbits, but they require a costly - from a computational point of view - codebook search. Multi-pulse algorithms, on the other hand, have to solve a nonlinear minimization problem for searching the pulse locations. Therefore, one problem of multipulse algorithms is the determination of optimal excitation sequences with a reasonable computational complexity. Such determination can be computationally simplified by using some approximations during the derivation, but this leads to suboptimal results. The classical multipulse algorithms described in [1, 2] in fact use sequential, step-by-step procedures where the next pulse location is determined assuming that all the other locations remain constant. The high performance of the algorithm described in [1, 2] is based on optimal adjustment of the pulse amplitudes during the location search. CELP is based on the assumption that the short- and long-term LPC residual can be modeled with a zero-mean Gaussian process. A stochastic codebook filled with instances of a Gaussian process is then exhaustively searched for the minimum of the mean-squared error. However, standard CELP [3] is not without pitfalls. First, we should assume that the codebook is large enough to model any kind of voiced and unvoiced excitations. Therefore, the CELP performance depends on the size of the codebook and on the assumed statistical distribution. Improvements can be obtained using training algorithms for the design of the random codebooks [4, 5] and using adaptive codebooks [6]. Second, exhaustive search imposes high computational complexity, which can be reduced using suitable structures of the codebook [7, 8]. Significant complexity reduction has been also described in [9–11], where a somehow different approach to low-bit rate coding is described.

A different approach to LP residual encoding is available from the year 2006, when Candès and Donoho proposed the Compressed Sensing or Compressed Sampling (CS) approach [12, 13]. CS approach states the possibility to sample a signal well below the Nyquist rate without degradation of the recovered signal provided that two hypothesis concerning sparsity and incoherence are satisfied. The approach moves the complexity from the encoder to the decoder. Thus the receiver requires much more computational power than the transmitter, since signal recovery at the receiver is performed by solving an optimization problem. Compressed sampling of LP residual is appealing since the residual is a sparse signal. Giacobello *et al.* in [14, 15] describe the computation of a sparse approximation of speech residual using compressed sensing.

In this paper we describe a different approach for the optimal determination of a sparse approximation of the residual sequence through a minimization carried out with simulated annealing. Standard multipulse finds pulse locations through a sub-optimal algorithm, trying to compensate the sub-optimality of pulse location with an optimal estimation of the pulses amplitudes. Our algorithm try to optimally estimate the sample locations while the amplitudes are the actual residual values at the found locations. Quality achieved and complexity figures indicate that the algorithm described in this paper is an alternative to Multipulse and Celp approaches. Instead of searching a codebook like Celp, the proposed algorithm determines the optimum residual samples using heuristic optimization. The algorithm has many interesting features. First of all, besides the initial LPC analysis and the computation of auto- and cross-correlations, it requires only additions during the optimum samples determination. Secondly, the convergence of simulated annealing applied to this problem is easily reached with a simple exponential cooling schedule. The algorithm's objective performance (segmental SNR) have been compared with the standard multipulse performance and with an adaptive compressed sampling algorithm. Moreover, informal subjective tests conducted with the proposed algorithm compared with standard CELP show indistinguishable quality at a bit-rate of $5-6$ Kbits. Finally, the algorithm can be implemented on a low-cost embedded system, as suggested by the simulation results.

The paper is structured as follows: in Section 2 the linear prediction of speech basis are briefly reviewed. In Section 3 we summarize the Compressed sensing algorithm to show its similarities with our approach. In Section 4 we briefly review the standard Multipulse approach which is compared with our approach. Section 5 describe our approach and Section 6 describes the optimization algorithm based on Simulated annealing. In Section 7 we report some experimental results about performance. In Section 7 we report also an analysis example of a speech frame. In Section 8 some concluding remarks and future word are discussed.

2 Preliminaries

According to the Linear Prediction (LP) of speech (e.g. [16]), in Z domain the LP residue is given by

$$E(z) = \left(1 - \sum_{i=1}^{p} a_i z^{-i}\right) X(z) = A(z)X(z) \tag{1}$$

where $E(z)$ is the LP residue, $X(z)$ is the speech signal, a_i are the linear prediction coefficients and p is the linear prediction order. Consequently, $X(z) = E(z)\frac{1}{A(z)}$. Turning to the discrete time domain, we have $x(n) = \sum_{i=0}^{p} h(i)e(n-i)$, where $x(n)$ is the speech signal, $h(i)$ is the impulse response of the all-pole system $\frac{1}{A(z)}$ and $e(n)$ is the LP residue. LP-based efficient coding of speech is based on finding an approximation $\tilde{e}(n)$ of the LP residual such that a high perceptual quality version $\tilde{x}(n)$ of the original speech signal is reconstructed at a lower bit rate using Equation (2).

$$\tilde{x}(n) = \sum_{i=0}^{p} h(i)\tilde{e}(n-i) \tag{2}$$

An important residual approximation approach is multipulse [1], where the residual is approximated with a series of pulses whose amplitudes are optimized. Other popular approaches are based on the selection of the optimum residue from a codebook of residues randomly generated [3]. Recently, Giacobello et al. [14] use Compressed sampling for representing the LP residual in a compressed sampling framework. In this paper we proposed to sample the residue by solving the related combinatorial problem with Simulated Annealing.

3 Compressed Sampling Principles

Candés *et al.* and Donoho in 2006 develop in [12], [13] the Compressed Sampling (or sensing) theory (CS). By CS theory, under sparsity and incoherence hypothesis, a signal can be reconstructed from very few samples, well below the Nyquist-Shannon rate. Sparsity means that a signal frame $\boldsymbol{x} = [x(1), x(2), \ldots, x(N)]$ may be expanded onto a basis $\boldsymbol{\Psi} = [\psi_1, \psi_2, \ldots, \psi_N]$ so that $x(n)$ is represented by only K significant coefficients, $K \ll N$. The expansion is represented by Equation (3) where only K coefficients in $\boldsymbol{c}$ are nonzero.

$$\boldsymbol{x} = \boldsymbol{\Psi c} \tag{3}$$

The signal is randomly sampled, so that $M < N$ random samples of x are taken, as described in Equation (4).

$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi x} = \boldsymbol{\Phi \Psi c} \tag{4}$$

The $M \times N$ measurement matrix $\boldsymbol{\Phi}$ is made by random ortho-basis vectors. By the CS theory, if $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are incoherent, the original signal $\boldsymbol{x}$ can be reconstructed from $\hat{x}$ within the approximation error ϵ by solving the optimization problem described in Equation (5).

$$\min_{c \in R^N} \sum_{n=1}^{N} |c(n)| \text{ such that } \sum_{i=1}^{N} [x(n) - \hat{x}(n)]^2 \leq \epsilon \tag{5}$$

Note that this optimization problem is solved at the receiver.

4 Review of LP Residue Approximation by standard Multipulse Approach

In the standard multipulse algorithm, [1], the LP residual is approximated with an impulsive sequence $u(n)$. The standard multipulse algorithm uses a closed-loop procedure for computing $u(n)$, once the impulse response $h(n)$ of the all-pole filter $\frac{1}{A(z)}$ is determined. The reconstructed signal $\widetilde{x(n)}$ is obtained by filtering the impulse excitation sequence $u(n)$ with the all-pole filter as reported in Equation (6).

$$\tilde{x}(n) = \sum_{i=0}^{p} h(i) u(n-i) \tag{6}$$

The pulse sequence $u(n)$ is a model of the prediction residual signal and is given by Equation (7) which is a linear combination of Kroneker delta functions.

$$u(n) = \sum_{i=0}^{M-1} \beta_i \delta(n - n_i) \tag{7}$$

In Equation (7) β_i, n_i, and M are the pulse amplitudes, the pulse locations, and the number of pulses respectively. In [1] the error between the input signal and the reconstructed one, $r(n) = x(n) - \tilde{x}(n)$, is weighted with a *perceptual* filter derived from $A(z)$. The perceptual filter is intended to un-emphasize the error energy in the high-energy regions of the signal spectrum, according to auditory masking criteria [17]. The best values of pulse amplitude and locations β_i and n_i is attained by minimizing the energy of the weighted reconstruction error. Because the weighting filter is linear, the minimization problem is equivalent to minimizing the un-weighted mean squared error between the weighted speech signal $x_w(n)$ and the corresponding weighted synthetic signal $\hat{x}_w(n)$. The optimization problem of [1] is described by Equation (8).

$$(n_i, \beta_i) = argmin \sum_{n=0}^{N-1} [x_w(n) - \hat{x}_w(n)]^2 \tag{8}$$

where N is the number of samples in the voice signal frame. The problem stated in Equation (8) can be solved at different levels of optimality depending on the procedure for the pulse location search. The algorithm described in [1] uses a step-by-step procedure for finding the M locations of the best pulse sequence. On the other hand, the sequential approach to location space scanning does not guarantee the optimal solution to the minimization problem.

5 A Novel Approach for Sparse Approximation of the Speech LP Residue

Our Sparse Approximation of the residual is simply the selection of a reduced number of residual samples that is still able to give good signal reconstruction performance. It is an approach of compressed sampling of the residual. Compressed sampling on N points sequence may be viewed as a combinatorial problem, in the sense that a small number of K samples are selected out of N to represent the sequence according to an optimization criterion. The number of the possible sets of K samples Γ is given by Equation (9).

$$\Gamma = \binom{\mathrm{N}}{\mathrm{K}} \tag{9}$$

In theory the selection of the optimal set of samples would be performed as follows. All the possible sets of compressed samples are generated, and for each set an error measure is computed. The set corresponding to the minimum error is thus selected. However, let us consider a simple example. If we start from a sequence of 20 ms at a sampling rate of $11K$ samples per second, we have a sequence of 220 points. If for example we want to select only 13 samples out of 220 points, the number of possible sets is about $3E^20$. Clearly, the computation of an error measure for each set is computationally impossible. For this reason we perform the compressed sampling with a Simulated Annealing optimization procedure [18].

The sampled residue $u(n)$ is given by Equation (10), where $e(n)$ is the LP residue, n_k are the sampling locations and M is the number of samples.

$$u(n) = \sum_{k=0}^{M-1} e(n_k)\delta(n - n_k) \tag{10}$$

The sampled residue is fed to the all-pole filter whose impulse response is $h(n)$. Thus the reconstructed signal $\hat{x}(n)$ is described by Equation (11).

$$\hat{x}(n) = \sum_{i=0}^{N} h(i)u(n-i) = \sum_{i=0}^{N} h(i) \sum_{k=0}^{M-1} e(n_k)\delta(n-i-n_k) = \sum_{k=0}^{M-1} e(n_k)h(n-n_k) \tag{11}$$

The reconstruction error E is reported in Equation (12).

$$\begin{aligned} E = &\sum_{n=0}^{N-1} [x(n) - \hat{x}(n)]^2 = \sum_{n=0}^{N-1} \left[x(n) - \sum_{k=0}^{M-1} e(n_k)h(n-n_k)\right]^2 = \\ &\sum_{n=0}^{N-1} x^2(n) + \sum_{i=0}^{N-1} \left[\sum_{i=0}^{M-1} e(n_i)h(n-n_i)\right] \left[\sum_{j=0}^{M-1} e(n_j)h(n-n_j)\right] - 2\sum_{n=0}^{N-1} x(n) \sum_{k=0}^{M-1} e(n_k)h(n-n_k) = \\ &\sum_{n=0}^{N-1} x^2(n) + \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} e(n_i)e(n_j) \sum_{n=0}^{N-1} h(n-n_i)h(n-n_j) - 2\sum_{k=0}^{M-1} e(n_k) \sum_{n=0}^{N-1} x(n)h(n-n_k) = \\ &\sum_{n=0}^{N-1} x^2(n) + \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} e(n_i)e(n_j)R_{hh}(n_i,n_j) - 2\sum_{k=1}^{M-1} e(n_k)R_{xh}(n_k) \end{aligned} \tag{12}$$

where $R_{hh}(n_i, n_j)$ is the auto-correlation of the impulse response at (n_i, n_j) and $R_{xh}(n_k)$ the cross-correlation at n_k.

5.1 The Simulated Annealing Algorithm

Simulated annealing (SA) is an heuristic optimization procedure proposed by Kirkpatrick in [19]. initially applied to the solution of combinatorial problems. Other reported applications of SA, besides combinatorial problems, include pin assignment optimization [20], image segmentation [21] and image reconstruction in Electrical Impedance tomography [22]. The compressed samples locations determination problem is indeed a combinatorial one as M locations out of N (the number of samples in a frame) must be found according to a suitable MSE criterion. In [20–22] Simulated Annealing optimization is applied only once. In the present applications, however, SA shall be applied for each signal segment. Therefore, real-time computation issue is fundamental.

Roughly, the concept of Simulated Annealing comes from the physical process called annealing, where where a solid is heated to melt. Subsequently the solid is slowly cooled to asses its reticule. In algorithmic terms, starting from an initial state, related to a starting temperature and associated to an energy definition, the state is randomly modified. If the energy associated to the new state is reduced with respect to the previous state, then the perturbed state is accepted and a new perturbation is performed. Otherwise, a new perturbation is performed. If the new state is in thermal equilibrium, as verified by the Boltzmann distribution, the perturbed state is accepted and the process continue. The sequence of random perturbations forms a Markov chain and it is generated until the thermal equilibrium is reached. Then, the temperature is reduced and another sequence of Markov chains is generated.

Thus, the algorithm consists in a sequence of Markov chains, each at a different decreasing temperature. The test for thermal equilibrium has been proposed by Metropolis *et al.* [23]. In summary, the Simulated Annealing algorithm is described in Algorithm 1.

Algorithm 1 Simulated Annealing Algorithm

(T_0, α)

procedure SA
    $T \leftarrow T_0$
    $E \leftarrow$ Energy definition
    Cooling Schedule $\leftarrow T = \alpha T$
    Current Model $\leftarrow$ random()
    while Not converged **do**
        New Model $\leftarrow$ *Randomly Perturbed Current Model*
        $\Delta_E \leftarrow$ *E(New Model) - E(Current Model)*
        if $\Delta_E \leq 0$ **then**
            Current Model $\leftarrow$ *New Model*
        else
            $r \leftarrow$ *randomnumber perturbation*
            if $(r \leq e^{-\frac{\Delta_E}{T}}))$ **then**
                Current Model $\leftarrow$ *New Model*
            end if
        end if
    end while
    return *Current Model*
end procedure

6 Compressed Sampling of Linear Prediction Residue

The SA solution of the sample selection problem is described as follows: a starting set of locations is randomly chosen and a generation scheme is suitably defined so that, given a set of locations, another set of locations is obtained. For each new set of locations a cost function E is computed as the weighted mean squared error between the original and the reconstructed signal. The new set of locations is accepted if the Metropolis test is satisfied. This process continues until a minimum is obtained. The Metropolis test is based on the Boltzmann distribution and uses ΔE and T_k. These two parameters are, respectively, the variation of the cost between two iterations, and the temperature that controls the annealing processe. The temperature T_k is updated at each iteration k.

6.1 Generation of Samples Locations

In theory, at each Simulated Annealing iteration the algorithm should manage a set of M different random locations. Recall that to each iteration corresponds a generation of a Markov chain where the M locations are randomly perturbed. Therefore, a multivariate random generator with a Gaussian distribution should be used. This approach would introduce this difficulty: each parameter might require a different annealing schedule for a good convergence and the computation of the cost function would be high. For these reasons, we developed a more robust samples locations generation scheme, where only one location at a time is modified according to a uni-variate Gaussian distribution. Therefore, the scheme we use to generate a new set of sample locations is the following:

1. Randomly select one pulse.
2. Perturb its location according to a uni-variate Gaussian distribution.

6.2 Algorithm for the determination of optimal residual samples

The initial value of the temperature, T_0, is set to the value of the standard deviation of the cost function computed during an initial free Markov chain. According to [18], the temperature update is realized as reported in Equation 13.

$$T_{k+1} = \alpha_k T_k, \quad \alpha_k = e^{-0.4 \frac{T_k}{\sigma^2}} \tag{13}$$

where σ^2 is the variance of the cost function during the k-th iteration.

The optimization algorithm for the determination of the optimal residual samples is described in the Algorithm 2.

The Metropolis test is based on the variation of the error function, as shown in Equation (14).

$$\begin{aligned} \Delta_E =& E_{new} - E_{old} = \\ & \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} e(n_i^{new}) e(n_j^{new}) R_{hh}(n_i^{new}, n_j^{new}) - 2 \sum_{k=1}^{M-1} e(n_k^{new}) R_{xh}(n_k^{new}) - \\ & - \left[\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} e(n_i^{old}) e(n_j^{new}) R_{hh}(n_i^{old}, n_j^{old}) - 2 \sum_{k=1}^{M-1} e(n_k^{old}) R_{xh}(n_k^{old}) \right] \end{aligned} \tag{14}$$

Algorithm 2 Compressed Sampling of the Linear Prediction Residue

```
(T0, α)
procedure SA1
    T ← T_0
    Current Locations set ← random()                ▷ random samples initialization
    while Not converged do
        while Iterations not terminated do
            New Locations set ← Randomly Perturbed Current Locations set
            Δ_E ← E(New) - E(Current)
            if Δ_E ≤ 0 then
                Current Samples Locations ← New Samples Locations
            else
                r ← random()
                if (r < e^(-Δ_E/T)) then                    ▷ Metropolis test
                    Current Locations set ← New Locations set
                end if
            end if
        end while
        T = αT                                      ▷ Temperature Annealing
    end while
    return Current Samples Locations
end procedure
```

However many of the terms before and after the minus sign in Equation (14) are equal because of the samples locations generation scheme: only one location is perturbed at each iteration. Thus many terms are canceled and Equation (14) reduces to Equation (15)

$$\begin{aligned} \Delta_E = & E_{new} - E_{old} = R_{hh}(n_{new}, n_{new}) - R_{hh}(n_{old}, n_{old}) - \\ & - 2e(n_{new}) \left[R_{xh}(n_{new}) - \sum_{k \neq old} e(n_k) R_{hh}(n_{new}, n_k) \right] + \\ & + 2e(m_{old}) \left[R_{xh}(n_{old}) - \sum_{k \neq old} e(n_k) R_{hh}(n_{old}, n_k) \right] \end{aligned} \tag{15}$$

where E_{new} and E_{old} are the error functions related to the actual and the previous iterations.

6.3 Convergence Behaviour

In Figure 1 the convergence behavior of the algorithm is shown. This figure describes the segmental SNR versus Markov chain length for 13, 16, and 22 samples per frame. Recall that in our experiments we use 20ms long frames, which at 11025 sampling rate means 220 samples. Thus 22 samples per frame is a 90% residual compression. The correct convergence is ensured by the saturation in the SNR as the Markov chain length increases. The best length value has been taken in correspondence of the saturation edge, and it was set to 300. The number of temperature decrements was experimentally found to be about 20 in order to reach the minimum.

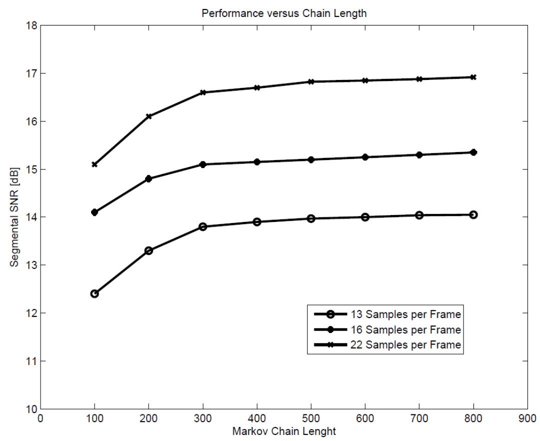

Fig. 1. Segmental SNR versus Markov chain length for various numbers of samples per frame.

All the products in Equations (15) are computed prior to the Simulated Annealing iterations. Thus each Simulated Annealing iteration require only additions and the iterations are very fast.

7 Experimental results

Experimental results are obtained with sentences extracted from the Artic speech dataset from CMU [24]. One hundred sentences spoken by two US male speakers are extracted from the database and the results are averaged among the two speakers. The data is down-sampled at 11000 samples per second, and analyzed with 20ms frames, namely 220 samples long. We used a 10th-order LPC analysis with a correlation approach. The segmental SNR is defined as shown in Equation (16).

$$Segmental\ SNR = \frac{10}{M} \sum_{m=0}^{M-1} log \frac{\sum_{n=N \cdot n}^{N \cdot n+N-1} x^2(n)}{sum_{n=N \cdot n}^{N \cdot n+N-1} [x(n) - \hat{x}(n))]^2} \quad (16)$$

The performance results, averaged over all the utterances and speakers, are shown in Fig.2. More precisely, Fig.2 shows the objective performances of the proposed algorithm and of the multipulse algorithms described in [1] versus the bit rate.

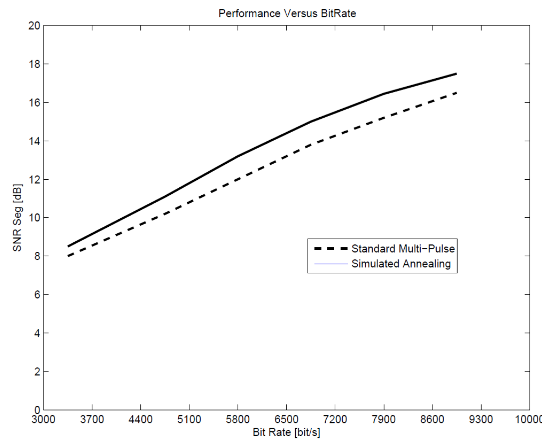

Fig. 2. Segmental SNR versus bit rate of the proposed algorithm compared to standard multipulse algorithm.

It should be noted that the bit rate used in this figure include the bits needed to code the excitation signal plus the 10-th order linear predictor, represented with Line Spectrum Pairs parameters and coded with 35 bits per frame. We should also note that no pitch prediction at all was used in obtaining all the data shown in Figure 2.

We then implemented in Matlab the adaptive compressed algorithm described in [25] and executed on the same data. The SNR results are reported in Figure 3, where the performance of our algorithm are included for comparison.

We finally report an example of an analysis performed with our algorithm on a frame of speech data. In Fig. 4 the input frame, 20ms long, is reported.

The corresponding 10-th order linear prediction residue is shown in Fig. 5.

The Simulated Annealing based optimization algorithm selects the 22 samples per frame reported in Figure 6, where the selected samples are overlapped with the residual signal. This compressed sampled residual is given in input to the $A(z)$ system to reconstruct the input signal.

The reconstructed signal is reported in Figure 7 using thin line, overlapped with the original signal which is plotted using a strong line in order to see the differences.

8 Final Remarks and Future work

In this paper, we describe an optimization algorithm based on Simulated Annealing for solving an optimization problem related to the approximate representation of the Linear prediction residual. We show that the approach turns out to be a form of compressed sampling of the residual signal. The algorithm described in this paper can produce coded speech at different levels of quality

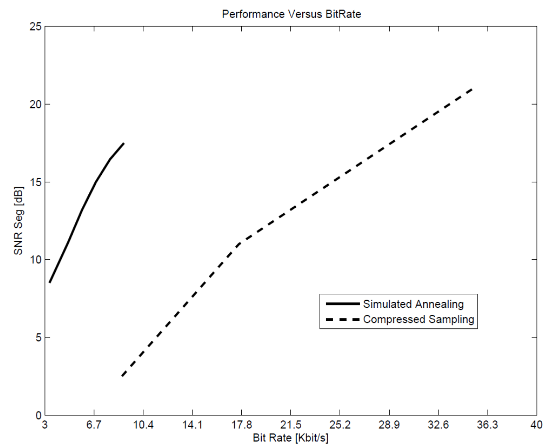

Fig. 3. Segmental SNR versus bit rate of the proposed algorithm compared to Compressed Sampling.

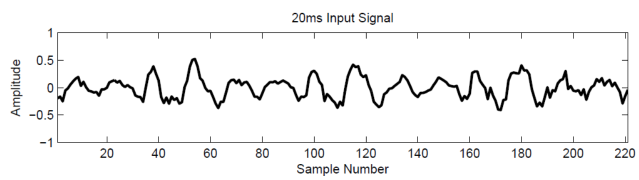

Fig. 4. Input 20ms speech frame sampled at 11000 samples per second.

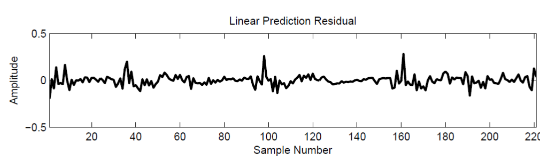

Fig. 5. 10-th order Linear Prediction residual.

and bit rate depending on how many linear prediction residual samples are used to represent the excitation signal. The optimization procedure requires only additions in every iteration. The LPC parameters used for coding were the LSP, which have been coded with 35 bits per frame. Future work concerns the real time implementation of the described algorithm on of the ARM-based embedded device currently available.

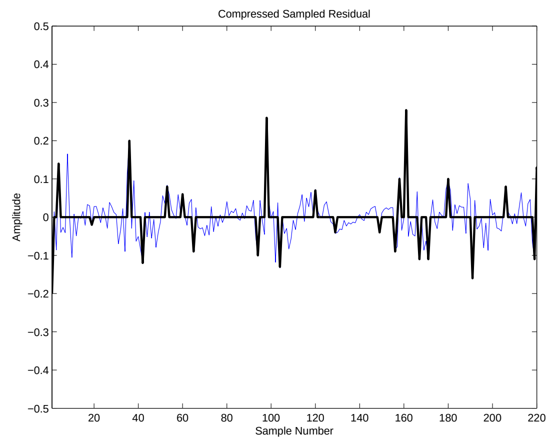

Fig. 6. Compressed sampling of the LP residual with 22 pulses per frame.

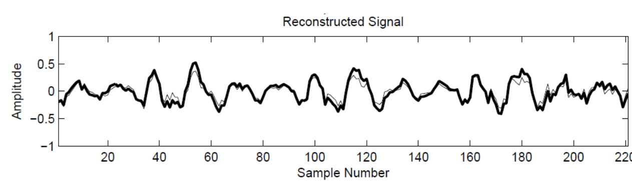

Fig. 7. Reconstructed signal with compressed LP residual (thin line) overlapped to the original signal (strong line).

References

1. Atal, B.S., Remde, J.R.: A new model of LPC excitation for producing natural-sounding speech at low bit rates. In: IEEE International Conference on Acoustics,

Speech, and Signal Processing, ICASSP '82, Paris, France, May 3-5, 1982. (1982) 614–617
2. Singhal, S., Atal, B.S.: Amplitude optimization and pitch prediction in multipulse coders. IEEE Trans. Acoustics, Speech, and Signal Processing **37**(3) (1989) 317–327
3. Schroeder, M., Atal, B.S.: Code-excited linear prediction(CELP): High-quality speech at very low bit rates. In: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85. Volume 10. (1985) 937–940
4. Sooraj, S., Anselam, A.S., Pillair, S.S.: Performance analysis of celp codec for gaussian and fixed codebooks. In: 2016 International Conference on Communication Systems and Networks (ComNet). (2016) 211–215
5. Manohar, K., Premanand, B.: Comparative study on vector quantization codebook generation algorithms for wideband speech coding. In: 2012 International Conference on Green Technologies (ICGT). (2012) 082–088
6. Stachurski, J.: Embedded celp with adaptive codebooks in enhancement layers and multi-layer gain optimization. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2009) 4133–4136
7. Kiran, C.G., Rajeev, K.: A fast adaptive codebook search method for speech coding. In: TENCON 2008 - 2008 IEEE Region 10 Conference. (2008) 1–4
8. Romaniuk, P.D.R.: Sparse signal modeling in a scalable celp coder. In: 21st European Signal Processing Conference (EUSIPCO 2013). (2013) 1–5
9. Sankar, M.S.A., Sathidevi, P.S.: Scalable low bit rate celp coder based on compressive sensing and vector quantization. In: 2016 IEEE Annual India Conference. (2016) 1–5
10. Chen, F.K., Chen, G.M., Jou, Y.D.: Complexity scalability design in coding of the adaptive codebook for itu-t g.729 speech coder. In: 2011 8th International Conference on Information, Communications and Signal Processing. (2011) 1–4
11. Ha, N.K.: A fast search method of algebraic codebook by reordering search sequence. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. (1999) 21–24
12. Candès, E.J., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans. Information Theory **52**(12) (2006) 5406–5425
13. Donoho, D.L.: Compressed sensing. IEEE Trans. Information Theory **52**(4) (2006) 1289–1306
14. Giacobello, D., Christensen, M.G., Murthi, M.N., Jensen, S.H., Moonen, M.: Retrieving sparse patterns using a compressed sensing framework: Applications to speech coding based on sparse linear prediction. IEEE Signal Process. Lett. **17**(1) (2010) 103–106
15. Giacobello, D., Christensen, M.G., Murthi, M.N., Jensen, S.H., Moonen, M.: Sparse linear prediction and its applications to speech processing. IEEE Trans. Audio, Speech & Language Processing **20**(5) (2012) 1644–1657
16. Atal, B.S., Hanauer, S.L.: Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. J. Acoust. Soc. Am. **50**(2) (1971) 637–655
17. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals and subjective error criteria. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '78, Tulsa, Oklahoma, USA, April 10-12, 1978. (1978) 573–576
18. Hamacher, H.W.: Theoretical and computational aspects of simulated annealing (P. j. m. van laarhoven e. h. l. aarts) and simulated annealing: Theory and applications (P. j. m. van laarhoven and e. h. l. aarts). SIAM Review **32**(3) (1990) 504–506

19. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. SCIENCE **220**(4598) (1983) 671–680
20. Li, Z.Y., Zhang, M.S., Long, Y.: Pin assignment optimization for large-scale high-pin-count bga packages using simulated annealing. IEEE Transactions on Components, Packaging and Manufacturing Technology **6**(10) (2016) 1465–1474
21. Yang, Y., Wang, Y.: Simulated annealing spectral clustering algorithm for image segmentation. Journal of Systems Engineering and Electronics **25**(3) (2014) 514–522
22. de Castro Martins, T., de Camargo, E.D.L.B., Lima, R.G., Amato, M.B.P., de Sales Guerra Tsuzuki, M.: Image reconstruction using interval simulated annealing in electrical impedance tomography. IEEE Transactions on Biomedical Engineering **59**(7) (2012) 1861–1870
23. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics **21** (1953) 1087–1092
24. : Cmu artic database. `http://www.festvox.org/cmu_arctic/index.html/`
25. Xu, Q., Yunyun, J.: Frame-based adaptive compressed sensing of speech signal. In: 7th International Conference on Wireless Communications, Networking and Mobile Computing. (2011) 1–4