Signal Processing

Improved Multipulse Algorithm for Speech Coding by Means of Adaptive Boltzmann Annealing

Enzo Mumolo, Alessio Rebelli
Dipartimento di Elettrotecnica, Elettronica ed Informatica, Università di Trieste
Via A.Valerio 10, 34127 Trieste - Italy
Giuseppe Riccardi
Dipartimento di Elettronica ed Informatica, Università di Padova
Via Gradenigo 6/A, 35131 Padova - Italy

Abstract. In this paper we will describe an algorithm, based on the Simulated Annealing technique, for solving the problem of multipulse excitation computation for speech coding applications. Multipulse excitation modeling is described by two set of parameters: pulse locations and amplitudes. The search for optimal pulse locations is the most burdensome task, since there is no closed form solution. Therefore, suboptimal approaches are used in order to maintain a good effectiveness in the parameter estimation while keeping the computational load within reasonable bounds. As compared to the one described in [2], the proposed algorithm achieves an effective improvement in objective performances of more than 1.5 dB. The price that must be paid for this performance improvement is the higher computational cost that this algorithm requires. The objective results of the proposed algorithm will be compared with the results of the algorithm described in [2], and the computational complexity will be analyzed.

1. INTRODUCTION

In this paper we will describe a novel algorithm for the computation of the multipulse sequence which excites the LPC synthesis filter in the signal reconstruction process. The multipulse approach, introduced by Atal [1] is able to generate high quality speech at a medium bit rate. The main problem related to multipulse sequence determination is to find the best sequence with a reasonable amount of computational complexity. Generally speaking it is possible to develop higher performance algorithms at the expense of an increase in computational complexity. An improved version of the original algorithm given in [1] is described in [2]. The improved version aims to obtain the excitation pulses with optimally adjusted pulse amplitudes.

Consider the combinatorial formulation of the multipulse sequence computation problem: in the present work this problem is solved by means of stochastic optimization procedures based on Simulated Annealing [3]; in particular, it will be shown that it is possible to effectively trade off computational load and objective performances. The segmental *SNR* (*SNR*SEG) shows a great improvement over the algorithm described in [2], that we take as a reference in the performance comparison. Moreover, we have derived a form of perturbation function which adapts itself to the actual value of the parameter that controls the annealing process.

So far, the applications of SA was intended for the solution of specialized problems where classical techniques offer poor performances, such as the design of particular filters [4] or the design of vector quantizers [5]. In such cases, however, the application of SA has to be done only once (i.e. "off line") and its high computational time is compensated by the good performances obtained. In our case, the application of SA is intended for real-time speech coding application; therefore a scheme to improve the algorithm efficiency is proposed. An interesting result of our application is that it can give an upper bound of the performances that can be obtained with a multipulse coder.

The paper is organized as follows: in section 2 the standard multipulse approach is briefly described, while in section 3 the SA solution is proposed. In section 4 the algorithm is described in details and an improved version, called SA/*L*, is introduced. Section 5 deals with the results obtained in terms of computational complexity and objective performance.

2. THE STANDARD MULTIPULSE TECHNIQUE

In the standard multipulse technique [2], the reconstruction process of a voice signal frame is achieved with

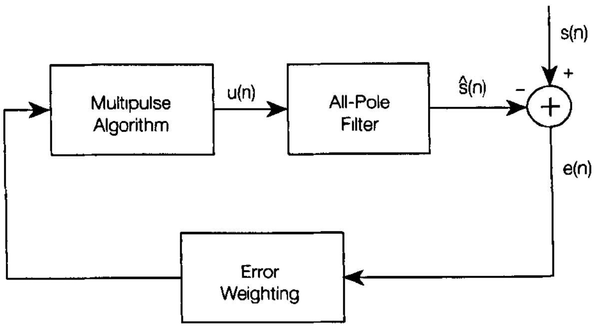

Fig. 1 - Analysis by synthesis scheme for multipulse sequence computation.

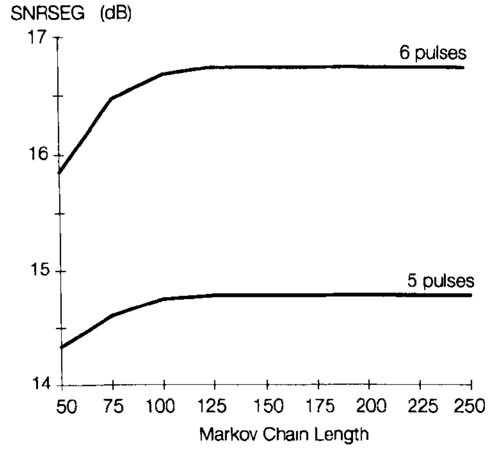

Fig. 2 - SNRSEG vs. the length of the Markov chains for the basic SA algorithm, for 5 and 6 pulses per subframe.

a pulse sequence which is fed into an all-pole filter, whose parameters are computed via classical LPC methods [6]. The multipulse algorithm, as shown in Fig.1, provides a closed loop procedure for computing the multipulse sequence $u(n)$, once the impulse response of the all-pole filter, $h(n)$, and the reference signal, $s(n)$, are given. The reconstructed signal $\hat{s}(n)$ is obtained by filtering the impulse excitation sequence $u(n)$ with the all-pole filter:

$$\hat{s}(n) = \sum_{i=0}^{n} h(i)u(n-i) + \hat{s}_0(n) \tag{1}$$

where $\hat{s}_0(n)$ is the contribution to $\hat{s}(n)$ from the filter memory and $h(n)$ is the impulse response of the all-pole filter. The pulse sequence $u(n)$ is a model for the prediction residual signal and it is given by a linear combination of Kronecker delta functions:

$$u(n) = \sum_{i=0}^{M-1} \beta_i \, \delta(n-n_i) \tag{2}$$

where β_i, n_i and M are the pulse amplitudes, the pulse locations and the pulse number, respectively. As depicted in Fig.1, the synthesis error $e(n) = s(n) - \hat{s}(n)$, between the reference signal and the synthesized signal, is weighted with a suitable "perceptual" filter $W(z) = A(z)/A(z/\gamma)$ derived from the LPC filter $A(z)$: this filter is intended to deemphasize the error energy in the high energy regions of the signal spectrum, according to auditory masking criteria [7]. The best excitation sequence is then attained by minimizing the energy of the weighted synthesis error, with respect to the parameters β_i and n_i. Since the weighting filter is linear, the minimization problem is equivalent to minimizing the unweighted mean squared error between a weighted speech signal $y(n)$ and the corresponding weighted synthetic signal $\hat{y}(n)$ as shown in [2, Fig.2]:

$$\min_{n_i, \beta_i} \sum_{n=0}^{N-1} \left[y(n) - \hat{y}(n)\right]^2 \tag{3}$$

where N is the number of samples in the voice signal frame. The problem stated in (3) leads to the well-known linear system:

$$\boldsymbol{R}\boldsymbol{\beta} = \boldsymbol{c} \tag{4}$$

where the generic element of matrix $\boldsymbol{R}$ is:

$$R(i,j) = \sum_{n=0}^{N-1} h_w(n-n_i) h_w(n-n_j) \qquad 0 \le i,j < M \tag{5}$$

and the generic element of vector $\boldsymbol{c}$ is:

$$c(i) = \sum_{n=0}^{N-1} h_w(n-n_i)\, y(n) \qquad 0 \le i < M \tag{6}$$

where $h_w(n)$ is the weighted pulse response. The solution of the linear system (4) depends on the choice of the procedure for the pulse location search. Both the algorithms in [1] and [2] use a step-by-step procedure for finding the M locations of the best pulse sequence. The high performances of the standard multipulse algorithm [2] is based on optimal adjustment of the pulse amplitudes during the location search. On the other hand, the sequential approach to location space scanning does not guarantee the optimal solution to the minimization problem. The main aim of this work is to consider an algorithm performing an unconstrained search in the pulse location space to find the best pulse sequence, while keeping optimized the intermediate pulse sequence at each evaluation point. As a whole, the tunability of the proposed algorithm parameters will also provide a break-even point between the computational load and the best solution looked for.

3. MULTIPULSE SEQUENCE OPTIMIZATION VIA SIMULATED ANNEALING

The multipulse sequence determination problem is a combinatorial one in the sense that M locations, out of N, must be found according to a suitable weighted MSE criterion. An exhaustive solution is normally not feasible, even with particular simplifications, as the ones described in [8] and [9]. A recently proposed optimization method, the Mean Field Annealing [10], may be very efficient when a mean-field theory is a good approximation to the stochastic cost function, under a quasi quadratic behaviour. Unfortunately, the multipulse problem is highly non linear and cannot be solved by MFA. A more suitable algorithm is Simulated Annealing (SA) [3]. The SA solution of the multipulse 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 is computed as the weighted mean squared error between original and reconstructed signal. The new set of locations is accepted if the cost function passes a so-called Metropolis test. This process continues until a minimum is obtained. The Metropolis test is based on the Boltzmann distribution exp $(-\Delta C/T_k)$, where C is the value of the cost function, ΔC is the variation of the cost between two iterations and T_k is a parameter that control the annealing process; it is usually called temperature and it is updated at each iteration k. If the probability of generating a set of configurations from a previous one depends on the iteration index the algorithm is called inhomogeneous because an inhomogeneous Markov chain is underlying. For inhomogeneous algorithms and under some assumptions it has been demonstrated that a sufficient condition for convergence is [3]:

$$T_k \le \frac{T_0}{\log k} \tag{7}$$

where T_0 is a constant and k the iteration index.

It has to be noted that the above condition is only sufficient. Thus, there may be an update schedule that does not satisfy that condition but still ensures convergence.

In each practical application of SA a sequence of inhomogeneous Markov chains is generated at decreasing values of the temperature. In these cases asymptotic convergence can only be approximated and therefore it is no longer guaranteed to reach the global optimum.

3.1. *The annealing process*

The annealing process is described by the following parameters:

- initial value of the temperature
- stop criterion
- length of the Markov chains
- the perturbation generation scheme.

These parameters will be described in the following subsections.

3.1.1. The initial value of the control parameter

The initial value of the control parameter T_0 is important for assuring a good convergence of the annealing process. In fact, if this value is too low, the minimization process will be probably driven to a local minimum, while if it is too high the first Markov chains would be useless because all the perturbations would be accepted. In practice, however, it is sufficient to find a correct order of magnitude. We used as T_0 the value of the standard deviation of the cost function C computed during an initial free Markov chain (i.e. a chain where every transition is accepted). Moreover, since speech is not stationary, the computation of T_0 must be done at each new frame.

3.1.2. Stop criterion and length of the Markov chains.

The stop criterion and the length of the Markov chains control the trade-off between the optimality of the obtained solution and the computational complexity. In fact the complexity is related to the number of iterations needed by the annealing process. The length of the Markov chains is related to the concept of quasi-equilibrium. From this point of view, the length of each chain should be such that a minimum number of perturbations is accepted. The drawback of this choice is that as the value of the control parameter decreases, the perturbations are accepted with a lower probability. Thus, the length would tend to infinite. Since for realtime applications, we need a constant number of iterations, we decided to use constant length Markov chains and a constant number of temperature decrements.

The number of iterations required by the Simulated Annealing for converging to the global minimun depends on the problem dimension: the higher the dimension of the problem, the greater the number of iterations of the algorithm.

For the search of optimal pulse locations in the multipulse excitation signal, the problem dimension is given by the following formula:

$$\mathrm{DIM} = \binom{N}{M} \tag{8}$$

where N is the frame length (expressed in samples) and M is the number of excitation pulses (per frame).

It follows that, if we fix N, the problem dimension is an explicit function of the number of pulses.

The stop criterion is established if we set the length of the Markov chains and the number of temperature decrements. The total number of iterations is clearly the product of these two parameters. Therefore the trade off between computational complexity and algorithm performance is not trivial since it requires extensive experimentation.

First, we deduced the Markov chain length value using a different stop criterion: the algorithm execution was forced to terminate when no more transitions were accepted. By fixing the number of pulses we made several tests, using twelve English phrases (six uttered by male and six by female speakers sampled at 8 kHz and with an average duration of about 2 seconds) with different chain lengths. The obtained results are reported in Fig. 2. This figure describes the Segmental *SNR* versus Markov chain length for 5 and 6 pulses and with N = 50. The correct convergence behaviour is insured by the saturation in the *SNR*Seg value as the Markov chain length increases; the best length value can be taken in correspondence of the saturation edge.

The best value of the number of temperature decrements is determined by applying the algorithm several times with the Markov chain length fixed as just described, and by averaging the number of temperature decrements the algorithm requires before stopping (the stop criterion we applied in these tests is the same as before).

In conclusion, on the basis of these results we set the length of the Markov chains to 100. The number of temperature decrements is, on the average, about 20. Thus the total number of iterations is, on the average, about 2000.

3.1.3. The perturbation generation scheme.

The standard simulated annealing may be characterized by a generating probability density, an update temperature schedule and an acceptance probability. If the cost function belongs to the class of Gaussian-Markovian systems, the generating probability should be chosen as a multivariate Gaussian distribution. It can be shown that, in this case, the acceptance probability has the form of a Boltzmann distribution.

For the same class of systems, a Cauchy update schedule has been proposed. In this case, the perturbations should be generated according to the Cauchy distribution, which has a fatter tail than the Gaussian one, permitting easier access to test local minima [11]. The sufficient condition for convergence, in this case, is faster: $T_k = T_0/k$. However, if the dimension of the configuration set is D, a D-dimensional Cauchy random generator should be used, and there is no fast algorithm for the computation of a D-dimensional Cauchy random generator.

In general a multivariate random generator should be used in the multipulse problem but it introduces the following problems: first, each parameter of the parameter set might require a different annealing schedule for a good convergence; secondly, the computation load of the cost function is quite high. For these reasons, we developed a more efficient scheme based on the modification of only one pulse at a time. The pulse locations perturbation scheme is therefore:

- select at random one pulse.
- perturb its location according to a suitable distribution.

This leads to a simpler and more efficient algorithm. The generation distribution function we used is a univariate Gaussian. Moreover, it is important to control the variance of the Gaussian distribution in such a way that, as the temperature decreases, a reduced subset of locations are tested by the annealing process. Then, the variance should be close to T_0 at the beginning and it should decrease as the process goes on. This observation led to the following generation distribution:

$$g(x, T_k) = \frac{1}{\sigma^2(T_k)\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2(T_k)}} \tag{9}$$

where

$$\sigma^2(T_k) = T_0 \cdot \left(1 - e^{-\frac{T_k}{2}}\right) \tag{10}$$

The constant 2 used in eq. (10) has been chosen empirically. This generation function adapts itself to the actual temperature value. The algorithm has been therefore called "Adaptive Boltzmann Annealing".

3.1.4. Temperature update.

We used the following temperature update rule [3]:

$$T_{k+1} = a_k \cdot T_k \tag{11}$$

where

$$\alpha_k = e^{-0.4\frac{T_k}{\sigma^2}} \tag{12}$$

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

This rule is reported, in solid line, in Fig. 3. The dashed line represents (7), the theoretical update rule related to the convergence of inhomogeneous algorithm.

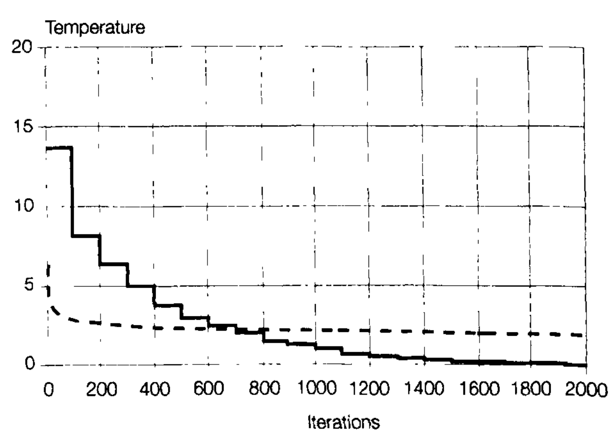

Fig. 3 - Temperature decrements vs. number of iterations (solid line). The dotted line shows the exponential law for the inhomogeneous SA algorithm convergence.

4. DESCRIPTION OF THE BASIC ALGORITHM

The weighted squared error between original and synthetic speech becomes minimum when the amplitudes are computed by solving (4), and is given by:

$$E_{\min} = \sum_{n=0}^{N-1} y^2(n) - \sum_{k=0}^{M-1} b_k\, c(k) \qquad (13)$$

Each iteration of SA, therefore, consists in the solution of (4) and the computation of the cost function through (13). Thus, a very high number of operations would be required. However, since only one pulse is modified at each iteration, only one row and one column of the matrix $\boldsymbol{R}$ are modified. Thus, efficient algorithms for the solution of (4) can be used [12]. The overall algorithm is described in Fig. 4.

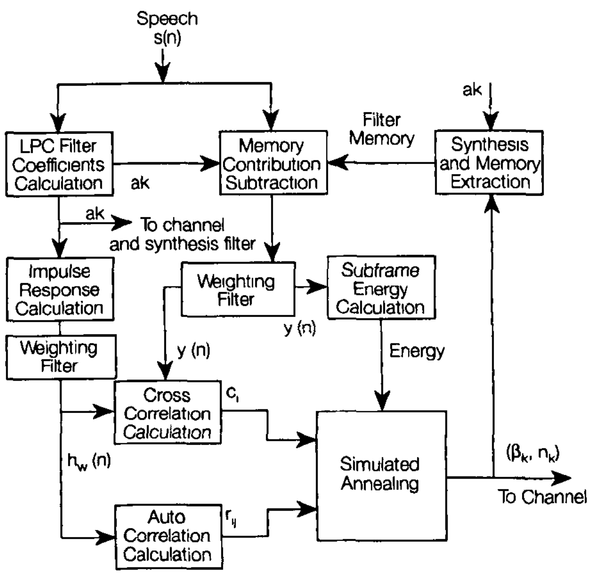

Fig. 4 - Overall block diagram of the SA algorithm.

4.1. *Improving efficiency*

The basic algorithm, described in the previous section, gives the best results but is the most complex. For practical applications we have to look for suboptimal solutions. One possibility is to maintain fixed the locations and amplitudes of all the pulses except the one whose location is being modified by the perturbation algorithm. This way, the complexity required by the solution of (4) decreases dramatically, because we need to solve the matrix equation with respect to a single row-column modification of the $\boldsymbol{R}$-matrix. Moreover, the computation of the mean squared error is greatly simplified when compared to (13). In fact, it will be shown that the error at the current iteration can be obtained by modifying the error at the previous iteration in a simple recursive way.

This way, however, the performance would be low. In order to partially overcome this problem, we compute an exact solution of (4) every L iterations. In the sequel this algorithm will be called SA/L and consists by the following steps:

a) modify, according to (9) one pulse location n_{ja} where j means that the j-th pulse has been selected and "a" means that the location is the actual one.
b) find the corresponding β_{ja} amplitude by solving with respect to the ja-th row of (4):

$$\beta_{ja} = \frac{c(ja) - \sum_{i,i\neq ja} R(i,ja)\cdot \beta_i}{R(ja,ja)} \qquad (14)$$

c) compute the current error E_a from previous error E_o.

The error to be minimized is:

$$E = \sum_n \left[y(n) - \hat{y}(n)\right]^2 = \sum_n \left[y(n) - \sum_k \beta_k\, h_w(n - n_k)\right]^2 \qquad (15)$$

Hence

$$E = \sum_n y^2(n) + \sum_{i=1}^{M}\sum_{j=1}^{M} \beta_i\, \beta_j\, R(i,j) - 2\sum_{k=1}^{M} \beta_k\, c(k) \qquad (16)$$

Since we modify only one pulse at each iteration, if we call E_o the error relative to the old configuration, it is easy to show that the relationship between E_o and E_a is the following:

$$\begin{aligned} E_a = {} & E_o + \beta_{jo}\cdot 2\cdot c(jo) - b_{ja}\cdot 2\cdot c(ja) + \\ & \beta_{ja}^2 \cdot R(ja,ja) - \beta_{jo}^2 \cdot R(jo,jo) - \\ & 2\cdot \beta_{jo} \sum_{i\neq jo} \beta_i \cdot R(jo,i) + 2\cdot \beta_{ja} \sum_{i\neq ja} \beta_i \cdot R(ja,i) \end{aligned} \qquad (17)$$

where n_{jo} is the old position of the j-th pulse within the frame and n_{ja} is the new one.

d) repeat the iterations from a)

e) every L of such iterations, solve (4) by Cholesky decomposition.

Fig. 5 describes the Segmental *SNR* versus Markov chain length for 5 and 6 pulses and with N = 50 for SA/10.

The saturation phenomenon, in this case, is less evident than in Fig.2 because of the greater problem dimension. In fact, in this case, the configuration set is determined by both the pulse positions and amplitudes and not only by the pulse positions as in the basic SA algorithm.

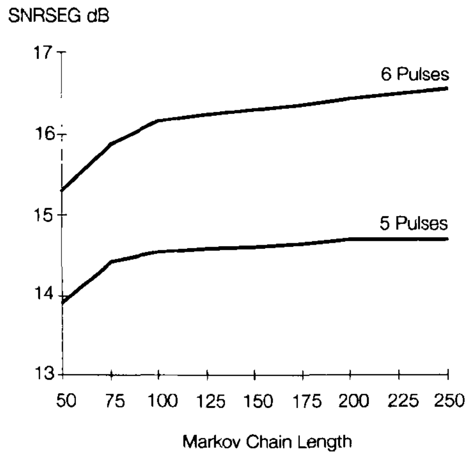

Fig. 5 - *SNR*SEG vs. the length of Markov chains for the SA/*L* algorithm for 5 and 6 pulses per subframe.

5. RESULTS AND DISCUSSION.

5.1. *Computational complexity*

The major problem related to the application of Simulated Annealing to the multipulse excitation calculation, is represented by the high computational complexity. In the following we will define the complexity in terms of number of multiplications and additions per second.

Since both the problem dimension and the number of operations per iteration depend on the number of pulses and on the frame length we divided the LPC analysis frame in more subframes.

Let us make the following additional definitions:

- ***CLP*** : Number of operations per frame for LPC parameters computation
- ***CSA*** : Number of operations per subframe for Multipulse sequence computation
- ***Fc*** : Sampling frequency (Hz)
- ***NSub*** : Number of subframes per frame
- ***NFSub***: Subframe length
- ***Nit*** : Total number of iterations

The total number of operations/s of the algorithm is given by:

$$CC = (CSA \cdot NSub + CLP) \cdot Fc/N \qquad (18)$$

where ***CLP*** is negligible with respect to ***CSA · NSub***.

Let us consider the basic algorithm. At every iteration we need to evaluate the new configuration cost, which means computation of pulse amplitudes and average squared error. To compute the error using (13) we need M additions and M products per subframe. To calculate the optimal pulse amplitudes, first we have to build the M by M matrix $\boldsymbol{R}$ and the M by 1 vector $\boldsymbol{c}$ and then we have to solve eq. (4).

The number of operations needed for the computation of the matrix elements is negligible with respect to the solution of (4), since all their possible values can be computed once, at the beginning of each frame.

As pointed out in section 4.1., the computation of (4) requires an unaffordable number of operations, even if efficient algorithms are used [12]. Let us then consider the improved algorithm. The pulse amplitudes are calculated by solving (4) in $\lfloor Nit/L + 1 \rfloor$ iterations, where $\lfloor \cdot \rfloor$ is the integer part of the argument. In the remaining iterations the amplitudes are calculated in the approximated way. For each iteration related to approximate solutions, the operations needed are: M multiplications and $M - 1$ additions for amplitude calculation with eq. (14) and $2M + 3$ multiplications and $2M + 2$ addititions for the computation of the cost function with eq. (17).

For the other iterations the number of multiplications is:

$$\frac{M^3 + 6M^2 - 7M}{6} \qquad (19)$$

and the number of additions is:

$$\frac{M^3 + 9M^2 - 10M}{6} \qquad (20)$$

respectively, for the exact solution of (4) and M multiplications and M additions for the computation of the cost function with eq. (13). Therefore, the total number of multiplications CSA_m and of additions CSA_a needed for the computation of the multipulse sequence is, for each subframe:

$$CSA_m = \left\lfloor \frac{Nit}{L+1} \right\rfloor \frac{M^3 + 6M^2 - M}{6} + \left[Nit - \left\lfloor \frac{Nit}{L+1} \right\rfloor \right] (3M + 3) \qquad (21)$$

and

$$CSA_a = \left\lfloor \frac{Nit}{L+1} \right\rfloor \frac{M^3 + 9M^2 - 4M}{6} + \left[Nit - \left\lfloor \frac{Nit}{L+1} \right\rfloor \right] (3M + 1) \qquad (22)$$

respectively. Let us consider the following typical case, which corresponds to 960 pulses/s:

$N = 100$
$NSub = 2$
$M = 6$
$L = 10$
$Nit = 2000$
$Fc = 8000$ Hz

In this case, we obtain $CSA_m = 51050$ and $CSA_a = 50127$, which is the number of multiplications and additions per subframe. Using (18), this means $8.1 \cdot 10^6$

multiplications and $8 \cdot 10^6$ additions per second. In Fig.6 the number of multiplications and additions per second for different values of M is reported (all the other parameters have been fixed to the above values). The number of operations to be added for LPC parameters computation and for the computation of the auto-correlation and cross-correlation elements is lower than 1 Mops/s.

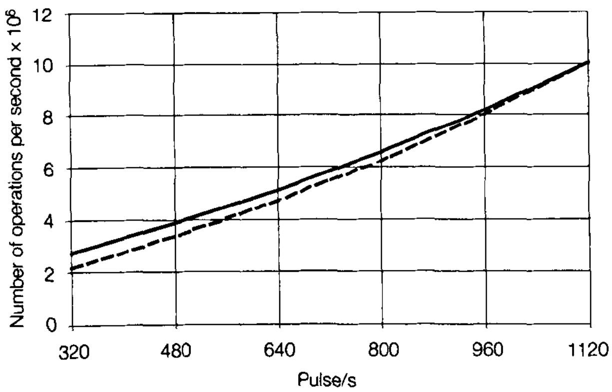

Fig. 6 - Number of multiplications (solid line) and additions (dotted line) vs. pulse rate for the SA/10 algorithm.

5.2. *Objective quality measurements*

In Fig.7 the segmental *SNR* measures are given as function of the pulse rate for the standard (Atal), the basic SA and the SA/L algorithm with $L = 10$ algorithms. By varying L a family of curves could be drawn. These plots have been obtained by processing the same data as described in section 3.1.2. It can be noted that the basic SA algorithm achieves, at a pulse rate of 1120 pulses per second, an improvement of more than 1.5 dB over the standard algorithm. At the same bit rate the improved SA algorithm, namely the SA/10, achieves an improvement of about 1 dB over the standard one. Furthermore, the higher the pulse density, the bigger the *SNR*SEG difference between the standard multipulse algorithm and the SA/L. Informal subjective tests have been conducted with different bit rates for both the SA/L algorithm, with $L = 10$, and for the standard one in order to compare them. The SA/L was judged to be always better than the standard one and the same behaviour as the objective measurements was reported. A pitch predictor has been also included in all the algorithms. Pitch prediction is important in multipulse coding schemes because it helps reduce the pitch dependence of multipulse excitation, especially for high-pitched speakers. A 1-tap pitch predictor yielded a *SNR*SEG improvement over the results described in Fig.7 of about 1.5 dB for the same data.

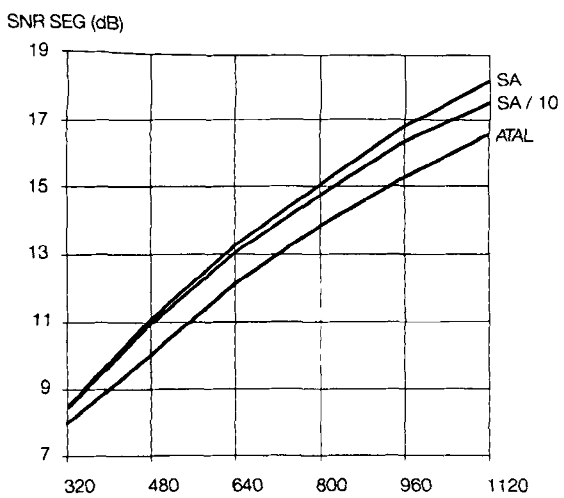

Fig. 7 - SNRSEG vs. pulse rate. The database consists of twelve sentences, six uttered by male and six by female speakers.

Finally, Fig.8 shows the *SNR* plot for the english sentence "Why were you away a year Roy?" uttered by a male speaker. It can be noted that the *SNR* improvement is consistent over the whole phrase.

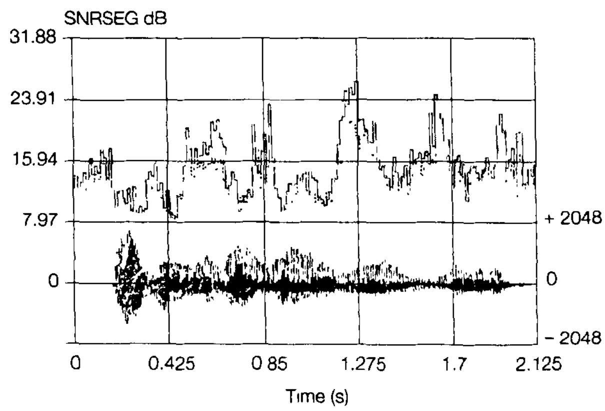

Fig. 8 - *SNR* frame-by-frame for the sentence "Why were you away a year, Roy?" uttered by a male speaker. The dotted line refers to the standard algorithm [2]; the solid line refers to the SA/10 algorithm. $Fc = 8$ kHz, pulse rate = 960 pulses/s.

6. CONCLUSIONS

The optimization of the multipulse sequence for modeling the LPC residual has been considered in this work. A stochastic optimization method, the Simulated Annealing algorithm, has been used to solve the multipulse minimization problem. The results show increased performance, as compared to the standard algorithm [2]. The proposed SA/L algorithm can reach different points of optimality, depending on the values of different parameters such as the length of the Markov chains, the number of temperature decrements, and the L-variable. Consequently, a trade-off between computational load and objective performance can be achieved. For this reason the algorithm is very versatile and can be of interest for applications in high quality speech coding.

Acknowledgement

This paper was partially supported by MURST 40%, "Algoritmi e strutture di calcolo".

Manuscript received on February 28, 1994.

REFERENCES

[1] B. S. Atal, J. R. Remde: *A new model of LPC excitation for producing natural sounding speech at low bit rates.* ICASSP 82 p.614-617.

[2] S. Singhal, B. S. Atal: *Amplitude optimization and pitch prediction in multipulse coders.* "IEEE Trans. on ASSP", Vol. 37, No. 33, March 89, p. 317-327.

[3] P. j. van Laarhoven, E. H. L. Aarts: *Simulated annealing: Theory and applications.* D.Reidel publishing company, 1987.

[4] N. Benvenuto, M. Marchesi, A. Uncini: *Applications of simulated annealing for the design of special digital filters.* "IEEE Trans. on ASSP", Vol. 40, No. 2, February 1992, p. 323-332.

[5] K. Zeger, J. Vaisey, A. Gersho: *Globally optimal vector quantizer design by stochastic relaxation.* "IEEE Trans. on ASSP", Vol. 40, No. 2, February 1992, p. 310-322.

[6] J. D. Markel, A. H. Gray Jr: *Linear prediction coding of speech.* Springer Verlag, N.Y. 1976.

[7] B. Atal, M. Schroeder: *Predictive coding of speech signals and subjective error criteria.* "IEEE Trans. on ASSP", Vol. 1, No. 3, June 1979, p. 247-254.

[8] M. J. Garside: *Some computational procedures for the best subset problem.* "Appl. Stat.", Vol. 20, 1971, p. 8-15.

[9] G. M. Furnival, R. W. Wilson: *Regression by leaps and bounds.* "Technometrics", Vol. 16, No. 4, 1974, p. 499-511.

[10] G. Bilbro et al.: *Optimization by mean field annealing.* In Advances in Neural Network Information Processing Systems, (D. Touretzky Ed.), Morgan-Kaufman, S. Mateo, CA, 1, p.91-98.

[11] H. Szu, R. Hartley: *Fast simulated annealing.* "Phys. Lett. A", Vol. 122, No. 3-4, p. 157-162.

[12] P. E. Gill, W. Murray: *Modification of matrix factorization after a rank-one change.* In State of the Art in Numerical Analysis, Academic Press, 1977, p. 55-82.