Abstract - In digital filter, when the coefficients assume power-of-two or a sum of power-of-two terms, the filter multiplier can be built using simple shift registers. In fact these filters are used in order to reduce both the VLSI silicon area and the computation time. In this paper power-of-two filters are extended to the case of adaptive filters. Due to the quantization and to the non uniform distribution in the coefficients domain, a classical gradient based approach, like LMS, cannot be successfully applied to perform the adaptation process. To overcome the problem of the filter adaptation a specific Tabu Search for time varying problems, suitable for real-time signal processing, is presented. Several experimental results demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

The use of adaptive systems, i.e. capable of altering their characteristics as the environment changes, is more and more present in applications, such as equalization of a communication channel or the automatic control of a system. Many adaptation procedures are based on the use of algorithms derived from the well known LMS [1]. In communication area, the growing need of high speed processing has attracted the attention of designers toward FIR digital filters with power-of-two, or combinations of power-of-two, coefficients; the latter allows to compute multiplications quickly and efficiently and to reduce the occupation of area on a silicon chip, thus achieving a higher degree of integrability. The tuning of such a device requires some modifications in order to make the employed algorithms: the solutions yielded by the LMS have to be rounded in order to be realized by a combination of two signed power-of-two terms, SPTs; several authors have attempted this way, such as [2] and [3], implementing original solutions.

On the other hand, these procedures greatly reduce the efficacy of the adaptation, because of the lack of controllability of the quantization error due to the computational method; the LMS, or one of its by-product, is not able to solve a combinatorial optimization problem which is the basis for the adaptation of filters with rounded coefficients.

Some researchers have been trying to use non conventional optimization procedures to tackle this kind of troubles, especially adopting a Genetic Algorithm [4]. The capability of representing a discrete domain, to which the solution belongs, leads to obtain a higher level of efficiency; the computational system performs an active control on the quantization process.

This paper deals with the use of the Tabu Search optimization algorithm [5]-[7] to drive the tuning of the coefficients, made of a sum of SPTs, of a communication channel equalizer. We have derived some functional conditions and have worked out some simple modifications in order to make the employed method capable to track the solution in a time varying environment; this is argued in the following section 2. In section 3 we indicate a brief formulation of the equalization problem and report our experimental results.

2. TABU SEARCH FOR TIME VARYING PROBLEMS

The Tabu Search, TS, algorithm has been used only in static optimization problems up to now; on the contrary, in adaptive applications the functional characteristics change as the time elapses. Referring to a constant domain, we have to solve the following formal problem:

$$\min f(x, t)$$

where \(f(\cdot)\) is the objective function, \(X\) denotes the feasible region and \(t\) explicitly indicates the temporal dependence. It is easy to perceive the solution, \(x^*\), is, in general, connected to the temporal parameter \(t\), \(x^*(t)\); the heuristic has to be able to track its movements within the search space.

In the next part of this section we will refer to the simple following version of the Tabu Search [5]:

0) Initialization of \(x, z, x^*, z^*, T\)
1) Generation of Candidate List,

$$CL=\{(m, mv) \mid m \in A \text{ and } mv=f(m(x))-f(x)\}$$
2) Selection of the best move, \(m=\arg \min CL\)
3) Execution of the best move, \(x\leftarrow m(x), z\leftarrow f(x)\)
4) Updating of the trial solution, \(x^*, z^*, \) and of the tabu set, \(T\)
5) If a stop criterion is not verified then go to step 1
6) The solution is \(x^*, z^*\).

where \(A\) is the set of admissible moves.

\(^1\) In this description the temporal parameter is omitted not to charge the notation.

\(^2\) The symbol \(\leftarrow\) denotes an assignment operation.
We will concentrate our attention on the characteristic memorization structure of the computational procedure; in particular, we will pay attention to the property of **temporal coherence**, that is the observance of the meaning of the actions in a time varying environment. We will examine three aspects of the algorithm: the generation of the Candidate List, CL, and the selection of the best move (steps 2 and 3), the admissibility mechanism and the solution updating (step 4).

Let’s consider a sequential generation of the CL and the changing of the objective function during this process due to time; the value of an admissible move, \( \mu \), is:

\[
mv(\mu) = f(\mu(x), t_1) - f(x, t_0)
\]

where the subscript 0 indicates the instant when the solution has been updated for the last time and \( \mu(x) \) denotes the application of the move \( \mu \) to the point \( x \). This quantity can be divided into two parts:

\[
mv(\mu) = [f(\mu(x), t_1) - f(x, t_1)] + [f(x, t_1) - f(x, t_0)]
\]

the former states the variation of the objective function due to the translation \( \mu \), the latter is due to time.

The move change can be kept under control by the heuristic, while the last term can be seen as an extern agent, whose displays are different from time to time. This aspect makes the comparison in step 3 senseless, because the selection is done between different objective functions. This effect is as much harmful as the temporal variations overlook the transferable ones. This reason leads us to ask the time to keep steady, at least during the execution of steps 2 and 3, so that the choice of the best move can be temporal coherent.

The second issue to analyze is the role of the short term memory, accomplished by the move status, in a non stationary environment. In the static case it is useful to avoid the entrapment in local minima, whereas with a dynamical objective function this task can even automatically be done by a simple translation of the cost function surface. The setting of the tabu tenure is less critical than in steady state, even though we have always to consider its constraining effect during the visit of the search space.

The last aspect of TS to survey is the storage of the objective function best value ever found during the execution of the heuristic, \( z^* \), in step 4.

In the time varying case this reference loses its meaning totally: the cost function used to compute it doesn’t exist any more. Therefore the adoption of this quantity as in the stationary case makes the algorithm temporal incoherent: the solution isn’t updated when it finds a better minimum than the previous one, but only if the value of the objective function is lower than that in the point \( x^* \). The new trial optimum should even be a maximum. The use of the quantity \( z^* \) puts an uncontrollable element into the TS, as much evident as the temporal variations are prominent.

The heavier fitting of the algorithm was done to face the latter drawback; roughly speaking, we execute more than a move keeping the time steady, i.e. the sequence of steps 1,2,3 is repeated for more times.

We thought of this modification considering the application we’ll do in section 3, which is related to a unimodal objective function, and the property of quick convergence when the TS deals with simple functions.

In order to render the adaptation procedure more flexible, the total number of repetitions of the sequence 1,2,3 is computed according to a measure of the environmental variations. Defining \( \delta \) as the relative difference between \( f(x,t) \) and the minimum of the cost function within a temporal window, we adopt the following rule:

\[
tot_{\text{it}}(\delta) = \begin{cases} 1, & \delta \leq 0 \\ [(\text{MAX}_{\text{IT}} - 1)\delta + 1, & 0 < \delta < 1 \\ \text{MAX}_{\text{IT}}, & \delta \geq 1 \end{cases}
\]

where \([\cdot]\) denotes the integral part and \( \text{MAX}_{\text{IT}} \) is set to 10. We also use a simple aspiration criterion within the TS mini-cycle, and the tabu tenure is set as half the total number of repetitions in order to prevent mean length loops.

### 3. Experimental Results

We have applied the algorithm described in the preceding section to the problem of linear equalization of a communication channel, using the RLS criterion [1],[8]. We have considered the following system:

\[
s[n] \quad \rightarrow \quad C(z) \quad \rightarrow \quad H(z) \quad \rightarrow \quad d[n]
\]

where \( \{s[n]\} \) is the transmitting binary sequence, \( C(z) \) states the equivalent transfer function of the communication channel [8], \( \{n[n]\} \) denotes the noisy sequence, \( y[n] \) indicates the output of the linear adaptive equalizer whose transfer function is \( H(z) \), \( d[n] \) is the target signal obtained delaying the sequence \( \{s[n]\} \) [1], and \( e[n] \) states the error which drives the TS to tune the coefficients of the linear FIR filter. We have chosen the classical RLS objective function to minimize:

\[
\varepsilon[n] = \sum_{k=1}^{\infty} \alpha^{n-k} e[k|n|^2] \\
e[k|n] = d[k] - x[k]^T \cdot w[n]
\]

where the inner product \( x[k]^T \cdot w[n] \) states the output of the equalizer with taps \( w[n] \) at instant \( k \); the vector \( x[k] \) and \( w[n] \) are defined as \( x[k] = x[k], x[k-1], \ldots, x[k-N] \).
For the simulations, we have considered an additive white gaussian noise $n[n]$, with zero mean, and two types of linear channels:

- $C_{\text{eqn}}(z) = 1 - 0.8z^{-1} + 0.5z^{-2}$
- $C_{\text{eqn}}(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$

and a non linear channel, $C_{\text{nlin}}$, built cascading the channel $C_{\text{eqn}}(z)$ and the non linear element $x+0.2x^2$.

We have employed equalizers of order $N=7$ and $N=10$ with coefficients belonging to the following domain:

$$D = \{ \alpha \in \mathbb{R} | \alpha = c_1 2^{-n} + c_2 2^{-h} \}$$

$\forall c_{1,2} \in \{-1,0,1\}, \forall g_{1,2} \in \{0,1,\ldots,B\}$

where $B$ has been set to 12 bits.

The tests have made up of two phases, an adaptation, to tune the equalizer coefficients by TS, and a measurement of the performances using the BER index over a sequence of five millions signal samples.

The results have been compared with those obtained using a quantized version of the Normalized LMS algorithm, Q-NLMS, where the filter taps are rounded to the nearer element belonging to $D$ after each iteration. They are plotted in the figures 1 through 6:

We get a higher and higher gain as the SNR increases; the Q-NLMS algorithm is not able to control the quantization error which is heavier and heavier. On the contrary, the TS can effectively contrast it, displaying just little different performances over the channel range.

A behavioural limit is also clear when we analyze on the base of the equalizer order [9]; it’s impossible to obtain lower BER levels using linear systems due to the inherent non linear nature of this kind of problem. We can attempt a further observation with respect of the on-line use of the proposed algorithm; the constraints we put, particularly the stop of the time during the steps 1,2,3, are very pressing from a computational point of view. This leads us to think about an actual batch solution, which will surely suffer of a drop in performances due to the delay it introduces.

### 4. CONCLUSIONS

In this paper we presented some modifications to the Tabu Search optimization algorithm to solve time varying problems. We tested the behaviour of the proposed method through the adaptive equalization problem of a communication channel using a FIR digital filter with combination of signed power-of-two coefficients; the results show a better quality of the adaptation process driven by the TS than that achieved by the Q-NLMS. The better capability in irregular domains of the Tabu Search is paid by a heavier computational effort which doesn’t allow an on-line use.

Our intention was to introduce non conventional optimization techniques for adaptive filtering problems; further developments can surely be done in order to work out modifications to the computational procedure and, mainly, the actual practical applicability.

### 5. REFERENCES

Figure 1 - Performance comparison. Channel $C_{\text{mp}}(z)$ and equalizer order 7.

Figure 2 - Performance comparison. Channel $C_{\text{mp}}(z)$ and equalizer order 10.

Figure 3 - Performance comparison. Channel $C_{\text{mp}}(z)$ and equalizer order 7.

Figure 4 - Performance comparison. Channel $C_{\text{mp}}(z)$ and equalizer order 10.

Figure 5 - Performance comparison. Nonlinear channel, $C_{\text{nl}}$, and equalizer order 7.

Figure 6 - Performance comparison. Nonlinear channel, $C_{\text{nl}}$, and equalizer order 10.