Genetic Algorithm Applied to Multipath Multiuser Channel Estimation in DS/CDMA Systems

Fernando Ciriaco and Taufik Abrão
State University of Londrina
Departamento de Engenharia Elétrica
86051-990, UEL, Brazil
Emails: fernandociriaco@sercomtel.com.br; taufik@uel.br

Paul Jean E. Jeszensky
Escola Politécnica of University of São Paulo
Departamento de Engenharia de Telecomunicações e Controle
05508-900, LCS-PTC-EPUSP, Brazil
Email: pjj@lcs.poli.usp.br

Abstract—This work analyses a heuristic algorithm based on the genetic evolution theory (GA) applied to multipath multiuser channel estimation (MuChE) in direct sequence code division multiple access (DS/CDMA). Using computer simulations we showed that the proposed GA-MuChE is capable of achieving a normalized mean squared error (MSE) in estimations of the order of 8% for slowly varying multipath fading channels ($f_d = 17 \text{Hz}$) and medium system load ($\approx 50\%$), with lower complexity than the maximum likelihood multiuser channel estimation (ML-MuChE) and gradient methods (Gradient Descent).

I. INTRODUCTION

In spite of existence of several works dealing with channel estimation, very few works analyze the channel estimation from the point of view of heuristic approaches. While heuristic are not perfect, i.e., they do not always find the optimal point, they are very efficient in attaining near-optimal solutions and significantly faster than conventional point-by-point exhaustive search techniques, especially in large solution spaces. Based on the ML rule, we developed a GA multiuser channel estimation technique that is capable to estimate the users’ complex channel coefficients from statistics provided by the bank of matched filters at the receiver. In [1] a genetic approach is proposed to jointly estimate the information bits and flat channel coefficients in a synchronous DS/CDMA system. In [2], a micro GA-based inter-symbol interference (ISI) channel estimation technique has been proposed, which employs the Viterbi algorithm for data detection in a single-user receiver over an AWGN channel. Differently of [1] and [2], this work considers multipath channels. Additionally, in [1] the genetic optimization process is done in the complex plan, while our paper considers binary mapping in the optimization process.

The majority of channel estimation works consider a multiuser or single-user maximum-likelihood approach with the multiuser signal correlation matrix inversion. The computational requirement of such optimization procedure is, however, prohibitively large when the number of users and paths increase. In practice, approximations are adopted. The heuristic approach for channel parameters estimates proposed in this work seeks to reduce the great computational complexity inherent to the ML method, maintaining the MSE of coefficients in acceptable levels. Based on the ML-MuChE approach, the authors of [3] derive channel estimates for long-code CDMA systems over multipath channels using training sequences. However, reliable estimates for the channel coefficients are obtained after processing an excessive frame length; considering static channels (without Doppler spread) the authors obtained a normalized MSE of 6% over a length frame of 100. On the other hand, considering the GA-MuChE heuristic approach, our simulation results have indicated a normalized MSE similar to the obtained in [3], however in slowly variant channel (instead of static channel) with a smaller frame length (around 10).

II. DS/CDMA SYSTEM MODEL

In a DS/CDMA system with binary phase-shift keying modulation (BPSK) shared by $K$ asynchronous users, the $k$-th user transmitted signal corresponding to an information sequence of length $I$ is given by:

$$x_k(t) = \sqrt{2P_k} \sum_{i=1}^{I} b_k^{(i)} s_k(t - iT_k) \cos(\omega_c t)$$

where $P_k = A_k^2/2$ represents the $k$-th user transmitted power; $b_k^{(i)} \in \{-1, +1\}$ is the $i$-th BPSK symbol with period $T_k$; $s_k$ is the signature sequence assigned to the $k$-th user, $\omega_c$ is the carrier frequency; $s_k(t) = \sum_{n=0}^{N-1} p(t - nT_c)e^{j\omega_c n}$ corresponds to the spreading sequence defined in the interval $[0, T_0]$ and zero outside, where $e_{k,n} \in \{-1, 1\}$ is the $n$-th chip of the sequence with length $N$ used by the $k$-th user; $T_c$ is the chip period and the processing gain, $\frac{T_0}{T_c}$, is equal to $N$; the pulse shaping $p(t)$ is assumed rectangular with unitary amplitude in the interval $[0, T_c)$ and zero outside.

Assuming that the signal $x_k(t)$ of each user propagates over $L$ independent slow Rayleigh fading paths, the baseband received signal in the base station is given by:

$$r(t) = \sum_{k=1}^{K} \sum_{T=1}^{L} w_{k,T} x_k(t - \tau_{k,T}) + \eta(t)$$

where $w_{k,T}$ and $\tau_{k,T}$ are the complex attenuation and the delay with respect to the timing reference at the receiver of the $T$-th path of the $k$-th user, respectively; the channel attenuations and delays are assumed to be constant during the estimation process; the random delay $\tau_{k,T}$ takes into account the asynchronous nature of the transmission, $d_k$, as well as the propagation delay, $\Delta_{k,T}$, for $k$-th user, $T$-th path, resulting in $\tau_{k,T} = \Delta_{k,T} + d_k$; $\eta(t)$ represents the AWGN. It is assumed that the $\ell$-path of the complex attenuation for the $k$-th user
over the \(i\)-th bit interval is \(w_{k,t}^{(i)} = \beta_{k,t}^{(i)} e^{j \phi_{k,t}^{(i)}}\), where the phase \(\phi_{k,t}\) has a uniform distribution over \(\phi_{k,t} \in [0, 2\pi)\) and the channel coefficients amplitude \(\beta_{k,t}\) represents the small scale-fading envelope following a Rayleigh distribution with probability density function \(f(\beta) = \frac{2\beta}{\sigma^2} e^{-\frac{\beta^2}{\sigma^2}}\), where \(\beta\) is the coefficients module and \(\sigma\) the multipath component average power \(\sigma = \mathbb{E}[\beta^2]\). Additionally, it is assumed that the channel gain is normalized for all users.

The received signal is discretized at the receiver by sampling \(r(t)\) at the chip rate. The observation vectors are formed by collecting \(N\) successive outputs of \(r(t)\). The observation vectors correspond to a time interval equal to one symbol period and start at an arbitrary timing reference at the receiver. Assuming all paths of all users are within one symbol period from the arbitrary timing reference, we will have only two symbols of each user in each observation window. Using vectorial notation equation (2) can be stated as [3]:

\[
r_i = \mathbf{U}^H \mathbf{b}_i + \mathbf{n}_i \tag{3}
\]

where \(r_i\) is the \(i\)-th \(N \times 1\) observation vector, \(\mathbf{U}\) is an \(N \times 2KN\) spreading matrix, \(\mathbf{Z}\) is a \(2KN \times 2KN\) channel response matrix, \(\mathbf{B}_i\) is a \(2KN \times 1\) channel response vector, and \(\mathbf{n}_i\) is a \(N \times 1\) complex Gaussian zero-mean random vector with independent elements each with variance equal to the bilateral power density, \(\sigma^2 = N_0/2\). The spreading matrix is constructed using the shifted versions of the spreading codes corresponding to the \(i\)-th and \((i+1)\)-th symbols of each user; for short codes results:

\[
\mathbf{U} = [\mathbf{U}_1^R \mathbf{U}_1^S \mathbf{U}_2^R \mathbf{U}_2^S \ldots \mathbf{U}_K^R \mathbf{U}_K^S] \tag{4}
\]

where \(\mathbf{U}_k^R\) and \(\mathbf{U}_k^S\) are defined in the top of this page. These matrix are constructed with the right and left parts of the spreading code of \(k\)-th user, respectively.

The channel response matrix is of the form \(\mathbf{Z} = \text{diag}(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K)\), where \(\mathbf{z}_k\) is the \(N \times 1\) channel response vector for the \(k\)-th user. For simplicity of analysis, we assume that the entire energy of \(t\)-th path is captured at \(\tau_{k,t} = q_{k,t} T_c\), i.e., at \(q_{k,t}\) position in \(\mathbf{z}_k\). For example, when user has two paths with delays \(\tau_{k,1} = 3T_c\) and \(\tau_{k,1} = 5T_c\), then:

\[
\mathbf{z}_k = \begin{bmatrix} 0 & 0 & 0 & w_{k,1} & 0 & w_{k,2} & 0 & \ldots & 0 \end{bmatrix}^\top \tag{5}
\]

Thus, the nonzero coefficients locations determine the path delays. Finally, the symbol vector has the form:

\[
\mathbf{b}_i = \begin{bmatrix} b_{1,i} & b_{1,i+1} & \ldots & b_{K,i} & b_{K,i+1} \end{bmatrix}^\top \tag{6}
\]

For channel estimation purpose, (3) can be rewritten as:

\[
r_i = \mathbf{U}^H \mathbf{B}_i \mathbf{z} + \mathbf{n}_i \tag{7}
\]

where the \(NK \times 1\) channel response vector is given by \(\mathbf{z} = [\mathbf{z}_1^\top \mathbf{z}_2^\top \ldots \mathbf{z}_K^\top]^\top\), and the \(2KN \times NK\) multiuser information matrix is defined in the top of this page.

Thus, we can estimate \(N\) channel parameters for each user. We have assumed that path delays for each user are within one symbol duration. Then, the number of nonzero coefficients in the effective channel response vector is determined by the number of paths and delays as in (5).

### III. ML Channel Estimation

The objective is to find the vector \(\mathbf{z}\) in (7) that maximizes the likelihood function of the channel response of all users using the knowledge of their spreading codes and transmitted bits. These bits could be available either as a preamble before the data or as bits in a separate pilot channel. In the estimation phase, training or pilot sequences use are assumed and in the tracking phase, data decisions from the detector are fed back to the estimator. The joint conditional distribution of \(I\) received observation vectors, given the knowledge of the spreading sequences, channel, and the information bits is:

\[
p(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_I | \mathbf{U}, \mathbf{Z}, \mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_I) = \exp \left\{ -\frac{1}{\sigma^2} \sum_{i=1}^I (\mathbf{r}_i - \mathbf{U}^H \mathbf{B}_i \mathbf{z})^H (\mathbf{r}_i - \mathbf{U}^H \mathbf{B}_i \mathbf{z}) \right\} \tag{8}
\]

The estimate that maximizes the likelihood function (ML) satisfies the equation [3]:

\[
\sum_{i=1}^I (\mathbf{U}^H \mathbf{B}_i)^H \hat{\mathbf{z}}_{ML}(I) = \sum_{i=1}^I (\mathbf{U}^H \mathbf{B}_i)^H \mathbf{r}_i \tag{9}
\]

Defining the \(NK \times NK\) multiuser signal correlation matrix \(\mathbf{R}_I = \frac{1}{I} \sum_{i=1}^I (\mathbf{U}^H \mathbf{B}_i)^H (\mathbf{U}^H \mathbf{B}_i)\) and the \(NK \times 1\) vector \(\mathbf{y}_I = \frac{1}{I} \sum_{i=1}^I (\mathbf{U}^H \mathbf{B}_i) \mathbf{r}_i\), the length of information sequence should be at least \(I = K + [K/N]\) for \(\mathbf{R}_I\) to be full rank.1 Therefore, assuming that \(\mathbf{R}_I\) is full rank, we can write the ML channel estimation as:

\[
\hat{\mathbf{z}}_{ML}(I) = \mathbf{R}_I^{-1} \mathbf{y}_I \tag{10}
\]

where \(\hat{\mathbf{z}}_{ML}(I)\) is jointly Gaussian with mean \(\mathbf{z}\) and covariance \(\frac{1}{I} \mathbf{R}_I^{-1}\) [3].

Next, we will develop a method based on heuristic approach to estimate the channel coefficients from few observations of the received vector using the knowledge of the spreading codes and transmitted bits.

1This is based on the assumption that random spreading codes are used and the spreading codes over this duration are linearly independent.
IV. GENETIC CHANNEL ESTIMATION ALGORITHM

Using heuristic algorithms, like genetic algorithm, we can compute the ML estimate recursively in an optimization subspace search. We apply genetic algorithm optimization techniques in the multiuser channel estimation problem (GA-MuChE). The motivation is that a direct computation of the exact ML channel estimate involves the computation of the correlation matrix and then the computation of the MA at the end of the preamble \( P \). The direct computation of the correlation matrix inverse is computationally intensive. Then we will approximate the ML solution by the heuristic method updates the channel estimate as the preamble is being received instead of waiting until the end of the preamble. The GA-MuChE algorithm is capable of estimating the vector \( \hat{z} \) (module and phase) through the minimization of mean squared error, where the candidate space, reducing substantially the complexity in relation to an exhaustive search technique. Thus, the cost function for the GA-MuChE algorithm, \( \phi(\hat{z}) \), a measure for the squared error, will be minimized considering a smaller search space than the total; the estimates for all channel coefficients during the \( m \)-th bit duration are obtained by:

\[
\hat{z}^{(m)}_{GA} = \min_{\hat{z} \in \mathbb{C}^{N \times K}} \phi(\hat{z})
\]

(11)

where, in this context, \( I \) is the processing window and the estimates \( \hat{z}^{(m)}_{GA} \) are obtained for each symbol period. This cost function is the negative of the log likelihood function (9), ignoring constants that are independent of the channel \( z \). Therefore, admitting that the set of initial bits (information bits matrix, \( B_i \)) and the spread sequences matrix (4) are known, the channel coefficients for all users can be simultaneously estimated. For sake of simplicity all users’ significant multipaths delays will be considered known.

The GA-MuChE total search universe will be characterized by all possible combination of \( \hat{z} \in \mathbb{C} \), resulting in a problem with infinite possibilities. Then it is necessary to work in a binary domain and to choose a certain quantization level, seeking to reduce the number of possible solutions. Thus, in order to increase the processing speed and offer certain implementation facility, the GA-MuChE optimization process is made just accomplishing operations with 1 and 0’s. Therefore, it is necessary accomplishing a mapping process (coding) of the problem from the complex domain to the binary domain.

The adopted mapping consists of separating \( \omega_{k,\ell} \) in real and imaginary parts:

\[
\omega^R_{k,\ell} = |\Re \{ \omega_{k,\ell} \}| \quad \text{and} \quad \omega^I_{k,\ell} = |\Im \{ \omega_{k,\ell} \}|
\]

(12)

where \( \Re \{ \} \) and \( \Im \{ \} \) represent the real and imaginary part operator of the complex number \( \{ \} \), respectively. After that, \( \omega^R_{k,\ell} \) and \( \omega^I_{k,\ell} \) are separated in integer and fractional parts:

\[
\chi^R_{k,\ell} = \lfloor \omega^R_{k,\ell} \rfloor \quad \text{and} \quad \psi^R_{k,\ell} = \omega^R_{k,\ell} - \chi^R_{k,\ell}
\]

\[
\chi^I_{k,\ell} = \lfloor \omega^I_{k,\ell} \rfloor \quad \text{and} \quad \psi^I_{k,\ell} = \omega^I_{k,\ell} - \chi^I_{k,\ell}
\]

where the operator \( \lfloor \cdot \rfloor \) returns the greatest integer not larger than \( n \). The values \( \chi^R_{k,\ell} \) and \( \psi^R_{k,\ell} \) are then digitizing through a analogue to digital converter (ADC),

\[
\hat{z}^R_{k,\ell} = \text{ADC } \left( \chi^R_{k,\ell} + \psi^R_{k,\ell} \right)
\]

\[
\hat{z}^I_{k,\ell} = \text{ADC } \left( \chi^I_{k,\ell} + \psi^I_{k,\ell} \right)
\]

where the operator ADC \([\cdot]_p\) returns the value of the argument into a binary vector with \( n \) bits and the notation \( \hat{z} \) represents binary version of \( \hat{z} \); the amounts of bits \( Q_{\text{int}} \) and \( Q_{\text{frac}} \) are input parameters of the algorithm and they contribute to determine the precision and complexity of the GA-MuChE. The values for the imaginary part are treated in similar form.

The basic processing unit (chromosome) is the vector formed by the integer and fractional parts of the real and imaginary part:

\[
\Lambda_{k,\ell} = \left[ \chi^R_{k,\ell} \chi^I_{k,\ell} \right]^{\top}
\]

(13)

Thus, the \( p \)-th individual is constituted by \( KL \) chromosomes, resulting in a binary column vector \( \hat{z}_p \) given by:

\[
\hat{z}_p = [\Lambda_{1,1} \Lambda_{1,2} \ldots \Lambda_{1,L} \Lambda_{2,1} \ldots \Lambda_{K,L}]^{\top}
\]

(14)

Therefore, each individual size will be proportional to the total number of users, paths, the resolution number of bits for the integer and fractional parts, resulting an individual size equals to \( Q_{\text{indiv}} = 2KL(Q_{\text{int}} + Q_{\text{frac}}) \) bits. This is the first individual characteristic, i.e., the absolute value of real and imaginary parts\(^2\). The second individual characteristic is the sign of each part of the complex number and can be represented by just one bit. Therefore, the second characteristic of the individual that have to be optimized is just equal to the total number of users and the amount of paths, resulting in \( Q_{\text{sign}} = 2KL \) bits. The GA-MuChE objective is to optimize these two characteristics together, looking for the minimization of the cost function \( \phi(\hat{z}) \) (11). This work uses an adapted equation in order to find the population size for the MuChE problem based on [4]:

\[
P = 10 \cdot 0.3454 \left( \sqrt{\pi (Q_{\text{indiv}} - 1) + 2} \right)
\]

(15)

This equation is calculated in the GA-MuChE initialization stage and maintained constant in all generations.

The estimate of the coefficients for the first symbol, first population, with dimension \( Q_{\text{indiv}} \times P \), is obtained randomly:

\[
\hat{Z} = \left[ \hat{z}_1 \hat{z}_2 \hat{z}_3 \ldots \hat{z}_P \right]
\]

(16)

However, for the coefficient estimates of the second symbol period, the first individual of the \( \hat{Z}^{(m)} \) population for the \( m \)-th coefficient consists of the best individual found in the previous symbol period, i.e., \( \hat{Z}^{(m-1)} \). The other \( P - 1 \) individuals of the first population are randomly generated.

In the MuChE problem context the attitude is measured through the squared error function (11) and it is directly responsible for the death or life of individuals. In this work, the selection process chooses the best \( M \) (mating pool size) individuals from the population \( P \) as the parents for the next generation. Consequently, the \( P - M \) individuals with low

\(^2\)Without considering the sign of each term.
fitness scores are removed for the reproduction stage. The mating pool size \(M\) should be selected in order to guarantee the convergence velocity and the final solution quality [5]. For the MuChE problem \(M = 0.2Q_{\text{Indiv}}\) was adopted.

For the GA-MuChE genetic operators we have adopted the uniform crossover [5] and the mutation based on noise:

\[
\text{new}_{\text{individual}} = \text{sign} \left( \text{individual} + \mathcal{N}(0, \sigma_{\text{mut}}^2) \right)
\]

where \(\mathcal{N}(0, \sigma_{\text{mut}}^2)\) represents a Gaussian distribution with standard deviation \(\sigma_{\text{mut}}\) and expectation zero. The standard deviation is strongly related with the mean mutation rate[6].

The replacement strategy used here is the global elitism, where only the best \(P\) individuals from the joint population of parents and offsprings are maintained for the next generation.

Finally, the optimization process for the \(m\)-th coefficients estimation is finished after a fixed number of generations \((G)\); the vector of coefficients in a binary form that minimizes (11) is selected, \(\hat{\mathbf{z}}_{\text{best}}\); then the coefficient estimates vector is obtained by digital to analogue conversion

\[
\hat{\mathbf{z}}_{\text{GA}} = \text{DAC} \left( \hat{\mathbf{z}}_{\text{best}} \right). 
\]

A pseudo-code for the GA-MuChE is described below.

**Input:** \(P, \hat{\mathbf{z}}, M, G\)

**Output:** \(\hat{\mathbf{z}}_{\text{best}}\)

1. Initialize first population \(\mathbf{z} = 0\);
2. \(\hat{\mathbf{z}} = \text{DAC} \left( \mathbf{z} \right)\);
3. Evaluate the fitness\((\mathbf{z})\);
4. While \(g < G\) then:
5. \(\mathbf{z}_{\text{selected}} = \text{Selection}(\mathbf{z}, M)\);
6. \(\mathbf{z}_{\text{cross}} = \text{Crossover}(\mathbf{z}_{\text{selected}});\)
7. \(\mathbf{z}_{\text{new}} = \text{Mutation}(\mathbf{z}_{\text{cross}});\)
8. \(\hat{\mathbf{z}}_{\text{best}} = \text{DAC} \left( \hat{\mathbf{z}}_{\text{new}} \right)\);
9. Evaluate the fitness\((\mathbf{z}_{\text{new}});\)
10. \(\hat{\mathbf{z}}_{\text{new}} = \text{Replacement}(\hat{\mathbf{z}} \cup \mathbf{z}_{\text{new}});\)
11. End.

V. NUMERICAL RESULTS

In all Monte-Carlo simulations we have adopted the following parameters: the spread sequences are selected as pseudo-noise (PN) with processing gain \(N = 16\); the number of active asynchronous users is \(K = 8\); two-paths slow Rayleigh channels with rays uniformly delayed in the interval \([0; N − 1]T_\text{c}\) and uniform power profile with \(\mathbb{E} \left[ \beta_k^2 \right] = \mathbb{E} \left[ \beta_{k,1}^2 \right] = 0.5\), \(\forall k\) and \(E_b/N_0 = 10dB\). All \(K\) users were considered with an uniformly distributed velocity in the interval \([0; v_{\text{max}}]\), resulting in a maximum Doppler frequency of \(f_D = 0.015\) for a carrier frequency of \(f_c = 1\) = 2 GHz. Table I synthesizes the main system and GA-MuChE parameters adopted in our simulations. The normalized MSE for the channel coefficients estimate during the \(m\)-th bit is defined by

\[
\zeta_{\text{GA}}(m) = \frac{\|\hat{\mathbf{z}}_{\text{GA}}(m) - \mathbf{z}(m)\|}{\|\mathbf{z}(m)\|}.
\]

Fig. 1 shows the normalized MSE (over 30 realizations) against the frame length \(I\) for the ML and Gradient-Descent channel estimates with \(\mu = 0.002\) from [3], and the proposed GA-MuChE algorithm (over 3 realizations), considering varying multipath fading channels \((f_D = 17Hz)\). The simulation results show that ML and GA-MuChE reach the same performance, in terms of normalized MSE, with \(\mathbb{E}[\zeta_{\text{GA}}(m)] \approx \mathbb{E}[\zeta_{\text{ML}}(m)] \approx 0.055\) for \(10 < I_{\text{GA}} < 20\) and \(40 < I_{\text{ML}} < 50\), respectively, while Gradient-Descent channel estimate reaches the minimum average squared error \(\approx 3 \times \mathbb{E}[\zeta_{\text{GA}}(m)]\) for \(50 < I_{GD} < 60\). Note that for realistic preamble lengths that are needed to get a normalized mean square of at least \(\mathbb{E}[\zeta] \leq 0.1\) (10%), the GA-MuChE algorithm can match the performance of the ML estimate with \(I = 10\), while spreading the computation over the entire length of the preamble, reducing the computational resources needed per bit.

VI. COMPUTATIONAL COMPLEXITY

The computational complexity of the GA-MuChE is evaluated in terms of operations number or flops - float point operations [7] during the \(m\)-th symbol interval and compared with the Gradient Descent channel estimate algorithm [3].
Table II expresses the number of operations for the $m$-th channel coefficients set estimates obtainment. Using the system's values and the GA-MuChE parameters, Tab. I, figure 4 shows the complexity behavior as a function of the processing gain. Note that for a load of 50%, the complexity of both algorithms is practically the same when $N \leq 170$; above this value, the GA-MuChE presents smaller computational complexity than the Gradient Descent algorithm.

![Fig. 2. Evolution of the normalized MSE for $K = 8$ users as a function of the number of generations $G$; it's considered MSE for the initial coefficients ($\hat{\theta}^{(1)}$, $\hat{\theta}^{(2)}$ and $\hat{\theta}^{(3)}$) and an intermediate coefficient ($\hat{\theta}^{(197)}$), for a group of 700 estimates, Fig. 3.](image)

**TABLE II**

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Number of Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Descent</td>
<td>$3K^2N^3 + N^2 {K^2 + K}$</td>
</tr>
<tr>
<td>GA-MuChE</td>
<td>$PG {N^2K + N + 9Q_{\text{int}} + 6Q_{\text{tag}} + 4KM + 2}$</td>
</tr>
</tbody>
</table>

**VII. CONCLUSIONS**

For slowly varying channel conditions, the GA-MuChE has presented great accuracy in the channel coefficients estimates, considering DS/CDMA systems with medium loading. The GA-MuChE optimization process is accomplished in the binary domain, being able to control the computational complexity and the resolution (variables $Q_{\text{int}}$ and $Q_{\text{tag}}$), resulting in a complexity × estimates accuracy trade-off that can follow the progress of the DSPs technologies.

The GA-MuChE reached an MSE similar to ML, with the advantage of an expressive reduction in the computational cost and estimates latency (smaller frame length). When compared to the Gradient Descent channel estimate algorithm the GA-MuChe has a smaller MSE and similar complexity for $N \leq 170$ and expressively smaller when $N >> 170$.

**REFERENCES**


![Fig. 3. Tracking performance over 700T, for two users; system with $K = 8$ users, $L = 2$, Doppler spread $f_D = 17$ Hz, $NFR = 0$ dB and the preamble length $I = 10$ bits. (a) user 3, path 1; (b) user 7, path 2.](image)

![Fig. 4. Computational complexity increasing as a function of the processing gain.](image)