RECURSIVE FINITE INTERVAL CONSTANT MODULUS ALGORITHM FOR BLIND EQUALIZATION

Antonín Heřmánek[1], *Phillip Regalia*[2]

[1]Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

[2]Institute National de Télécommunication
Dept. Communication, Images and Information Processing, Evry, France

Abstract

In the paper, we reexamine the Finite Interval Constant Modulus Algorithm (FI-CMA) proposed by P. Regalia and we present our modifications leading to its recursive form. The proposed modifications are based on recursive updates of QR decomposition. The presented results shows the computation savings and ability to use Recursive FI-CMA for tracking time variant channels.

1 Introduction

In modern digital communication systems the main part of the receiver consumes an estimator of transmited symbols. These estimators consist typicaly on an equalizer and decision device. Receant systems (for expample GSM wireless system) use well known methods with training sequence where a part of signal is a priory known and frequently repeated. The equalizer search for that signal and adapt its parameters to minimize some criteria (typicaly MSE). Unfortunately this known sequence is unnegligable part of the overall message (abouve 18% in GSM). For that reasons the research effort of the last years were investigated to deconvolution algorithms working blindly i.e. without training sequence. Very popular blind algorithm called Constant Modulus Algorithm (CMA) was originaly proposed by Godard. In this section we review the Finite-Interval CMA proposed by P. A. Regalia in [2]. In the following text we consider that all vectors are column vectors.

System model: The symbol sequence to be transmited , $\{s_n\}$, is assumed to be independent and indenticaly distribured with non Gaussian probability distribution function and of constant modulus (CM). CM property of transmited symbols is in telecommunication easily done by digital modulation as is for example PSK. The data symbols are send throught Signgle-Input Multiple-Output (SIMO) discreat channel with impulse response matrix $\mathbf{H}$. We assume the channle is finite in the length and Bounded Input Bounded Output (BIBO) stable. Than the received signal has the form:

$$\mathbf{u}_n = \mathbf{H}\mathbf{s}_n + \mathbf{w}_n \tag{1}$$

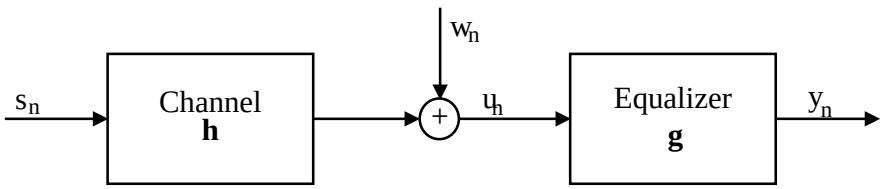

Figure 1: Discrete system model

where $\mathbf{s}_n = [s_n\ s_{n-1}\ \ldots\ s_{n-M}]^T$ is M last successive input symbols, $\mathbf{w}_n$ is additive white Gaussian noise vector, $\mathbf{H}$ is $P \times M_c$ channel impulse response matrix and P is number of antennas or oversampling factor. An equalizer is viewed as linear predictor of the order M and its output can be written in the form:

$$y_n = \sum_{k=0}^{M} \mathbf{g}_k^T \mathbf{u}_{n-k} = \mathbf{g}^T \mathbf{U}_n \tag{2}$$

where $\mathbf{g}$ and $\mathbf{U}$ are defined as :

$$\mathbf{g} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_M \end{bmatrix}, \qquad \mathbf{U}_n = \begin{bmatrix} \mathbf{u}_n \\ \mathbf{u}_{n-1} \\ \vdots \\ \mathbf{u}_{n-M} \end{bmatrix}$$

From the equation (2), N successive equalizer outputs can be directly rewritten in the matrix form as:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \\ \vdots \\ \mathbf{U}_N^T \end{bmatrix}}_{\mathcal{U}} \mathbf{g} = \mathbf{QRg} = \mathbf{Qw} \tag{3}$$

where the QR-decomposition of matrix $\mathcal{U}$ was used to obtain an orthonormal matrix $\mathbf{Q}$. Partitioning $\mathbf{Q}$ matrix row-wise, the current equalizer output can be expressed using (2) and (3) as:

$$y_n = \mathbf{q}_n^T \mathbf{w} \qquad \text{where} \qquad \mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_N^T \end{bmatrix} \tag{4}$$

Criterium: CMA tries to minimize a cost function defined by the constant modulus (CM) criterion which penalize deviation in the magnitude of the equalizer output from a fixed value. This criterion has the form:

$$J_{CMA}(g) = \frac{1}{4} E\left[\left(|y_n|^2 - \gamma\right)^2\right] \tag{5}$$

where $E[\cdot]$ is expectation operator and γ is a constant chosen as a function of the source alphabet. The FI-CMA is derived version of (5) where expectation operator is replaced by sumation over finite data interval and is defined as:

$$J(g) = \sum_{n=1}^{N} \left(|y_n|^2 - 1\right)^2 = \sum_{n=1}^{N} \left(\left|\mathbf{g}^T \mathbf{U}_n\right|^2 - 1\right)^2 \tag{6}$$

where constant γ was replaced by one without lost of generality because it doesnot change the position of the local extrema points. The equalizer coefficient vector $\mathbf{g}$ can be divided to the radial term ρ and direction term $\overline{\mathbf{g}}$ as follows:

$$\mathbf{g} = \rho \overline{\mathbf{g}} \qquad \|\overline{\mathbf{g}}\| = 1, \quad \rho = \|\mathbf{g}\|$$

Searching for local extrems of (6) with respect to radial therm ρ, we optain optimal ρ which minimize cost function (6) to be equal to $\rho_{opt}^2 = \frac{\sum y_n^2}{\sum y_n^4}$. With such as ρ the minimization of (6)

with respect to direction therm $\overline{\mathbf{g}}$ is equivalent to minimize the criteria defined as follow (for details see [2]):

$$\mathbf{F}(\mathbf{g}) = \frac{\sum_{n=1}^{N} y_n^4}{(\sum_{n=1}^{N} y_n^2)^2} = \frac{m_4}{m_2^2} \tag{7}$$

Rewriting equation (7) using (3) and with the $\mathbf{Q}$ orthonormal and with constraint $\|\mathbf{w}\| = 1$ the nominator sumation in (7) simplifies to $\sum y_n^2 = \mathbf{gUU}^T\mathbf{g}^T = \mathbf{wQQ}^T\mathbf{w}^T = 1$ and the criterion gets the form:

$$\mathbf{F}(\mathbf{g}) = \frac{\sum_{n=1}^{N} y_n^4}{(\sum_{n=1}^{N} y_n^2)^2} = m_4 \tag{8}$$

In [2], the minimum of (7) is reached by the steep-descent algorithm of the form:

$$\begin{array}{rcl} \mathbf{v}_{i+1} & = & \mathbf{w}_i - \mu(\mathbf{Q}^T\mathbf{y}^3 - \mathbf{F}_i\mathbf{w}_i) \\ \mathbf{w}_{i+1} & = & \mathbf{v}_{i+1}/\|\mathbf{v}_{i+1}\| \end{array} \tag{9}$$

with $\mathbf{F}_i$ defined by (8) and with step size approximation equal to $\mu = \alpha/\mathbf{F}_i$ where α is in the range $\langle 1/3\,,\, 2/3\rangle$. A small variation of that algorithm were experimentaly developed by P.A. Regalia of the form:

$$\begin{array}{rcl} \mathbf{v}_{i+1} & = & \mathbf{w}_i - \mu\mathbf{Q}^T\mathbf{y}^3/\mathbf{F}_i \\ \mathbf{w}_{i+1} & = & \mathbf{v}_{i+1}/\|\mathbf{v}_{i+1}\| \end{array} \tag{10}$$

with the optimal step size $\mu = \sqrt{N}$, where N is a block length. Variant (10) of FI-CMA has faster convergence than (9) and in sequel, proposed modifications are tested on this version. Note that minimal block size which enables successive blind deconvolution for CMA criterion is $N = M^2$.

Note : The presented FI-CMA algorithms work in batch mode, where several iterations of (9) (resp. (10)) are calculated for deconvolution of data block of length N.

2 Algorithm modifications

Because of the batch mode of the above algorithms, they can be used only for time-invariant channels or slowly variing channels with respect to block length. More over the solutions for two successive blocks do not have to correspond to the same system delay, the blocks must overlap and the search of the beginig of the message in the frame of data have to be donne for each block. Unfortunately, bigger block size N leads to higher computation complexity of deconvolution procedure and lowering N leads to frequent use of the "search of message beginning" procedure. The aim of the proposed modifications is to:

- use the above algorithms in recusrive (non-block) mode
- reduce the number of operation for long data streams
- develop an algorithm for time varying channels

Lets define the matrix $\mathcal{U}_n$ as in (3) with only the last N input data vectors $\mathbf{U}_n$:

$$\mathcal{U}_n = \begin{bmatrix} \mathbf{U}_{n-N+1}^T \\ \mathbf{U}_{n-N+2}^T \\ \vdots \\ \mathbf{U}_{n-1}^T \\ \mathbf{U}_n^T \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{n-N+1}^T \\ \mathcal{U}_n^- \end{bmatrix}, \tag{11}$$

$$\mathcal{U}_{n+1} \quad = \quad \begin{bmatrix} \mathbf{U}_{n-N+2}^T \\ \vdots \\ \mathbf{U}_n^T \\ \mathbf{U}_{n+1}^T \end{bmatrix} = \begin{bmatrix} \mathcal{U}_n^- \\ \mathbf{U}_{n+1}^T \end{bmatrix} \tag{12}$$

From (11) and (12) it is obvious that matrics $\mathcal{U}_n$ and $\mathcal{U}_{n+1}$ are similar, both share part $\mathcal{U}_n^-$. So $\mathcal{U}_{n+1}$ can be viewd as shifted version of $\mathcal{U}_n$ with first row deleted and with an additional row $\mathbf{U}_{n+1}$ at the bottom. Because of that structure of the matrics $\mathcal{U}_n$ and $\mathcal{U}_{n+1}$ we can apply append/delete row procedures of QR deconposition. These algorithms are described in [1] chapter 12.5.3 and its computation complexity is expressed in Table 1. The above row-wise decomposition of matrix $\mathcal{U}_n$, one update step consist of adding new row at the bottom and deleting the first row of R. Note than for deleting of a row the full matrix Q is needed where as for adding a row it is not.

The matrix $\mathcal{U}_n$ can be also viewed in the column wise decomposition. Define input data matrix $\hat{\mathbf{U}}_n \in \mathbf{R}^{N \times P}$ as :

$$\hat{\mathbf{U}}_n = [\mathbf{u}_{n-N} \ \ldots \ \mathbf{u}_{n-1} \ \mathbf{u}_n]^T$$

then the matrix $\mathcal{U}_n$ has the following structure:

$$\mathcal{U}_n = \left[\hat{\mathbf{U}}_n \ \hat{\mathbf{U}}_{n-1} \ \ldots \ \hat{\mathbf{U}}_{n-N} \right] \tag{13}$$

Again the data matrics $\mathcal{U}_n$ and $\mathcal{U}_{n+1}$ share the same submatrix and $\mathcal{U}_{n+1}$ can be viewed as left-to-right shifted version of the matrix $\mathcal{U}_n$ with P columns added to its left part and P columns deleted in its right part. The QR matrix can be updated by the appending/deleting a column update algorithms. These algorithms are described in [1] chapter 12.5.2 and its complexity is expressed in Table 1. Note that deleting the most right columns does need any computations and algorithm can work just with the first $P \cdot M$ columns of matrix $\mathbf{Q}$.

After QR actualization, one iteration of steep-descent algorithm (9) or (10) is proceed. Since QR actualization change also the matrix $\mathbf{R}$ and because $\mathbf{w} = \mathbf{Rg}$, after each iteration of (9) resp. (10) the equalizer coefficients $\mathbf{g} = \mathbf{R}^{-1}\mathbf{w}$ must be updated and after each QR actualization $\mathbf{w} = \mathbf{RG}$ must be back calculated. The proposed algorithms consist of these steps:

1. initialization:
$$\begin{aligned} \mathbf{Q} &= eye(N, N) \\ \mathbf{R} &= eye(P \times N) \end{aligned}$$
2. actualization of QR by append/delete row/column routine (as in [1])
3. calculate $\mathbf{w} = \mathbf{Rg}$
4. calculate one or few iterations of the steep-descent procedure (9) or (10)
5. calculate back $\mathbf{g} = \mathbf{R}^{-1}\mathbf{w}$
6. repeat steps $2. - 5.$ until end of data stream

Note: We expect that $\mathbf{R}$ does not change signnificantly so $\mathbf{g}$ do not have to be calculated and steps 3. and 5. can be skipped. It results to a bit slower convergency but brings significantly lower computation complexity.

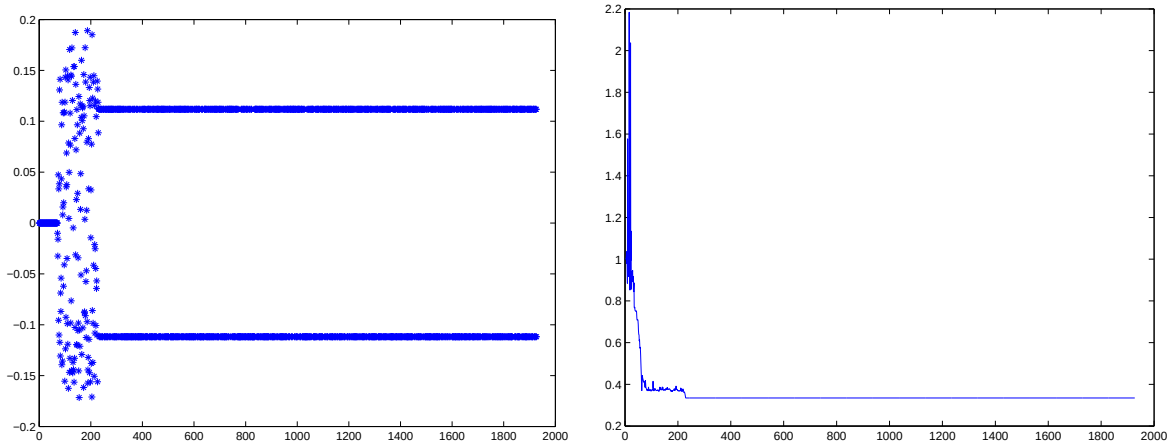

Figure 2: Equalizer output and evolution of cost function $\mathbf{F}$ for P=2,M=9 with no noise and static $\mathbf{H}$

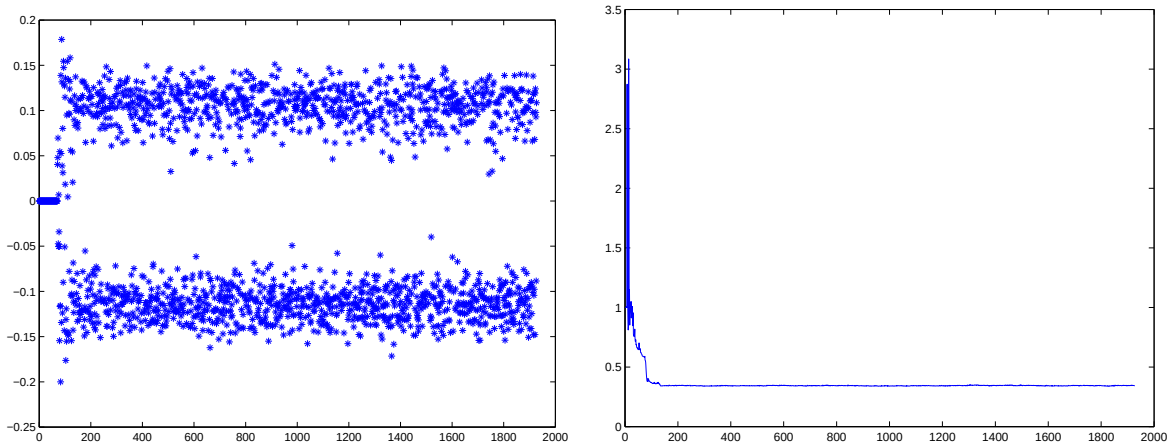

Figure 3: Equalizer output and evolution of cost function $\mathbf{F}$ for P=2,M=9 with additive Gaussian white noise 14db and static $\mathbf{H}$

3 Simulation results

We have consider a system with single input/two outputs channel of the 9^{th} order and linear equzlizer of length 9. The input data stream $\mathbf{s}_n$ is a random binary sequence of the length 2000. In some cases the white Gaussian noise with SNR 14db is added to the channel output. Figures 2 and 3 show equalizer output and cost function evolution for the time invariant channel and figures 4, and 5 show the results for the time variant channel. The evolution of the channel parameters for time varying case is shown in Figure 7 (channel impulse response change just in two coffitients).

For comparison of proposed algorithm with original FI-CMA, the equalizer output and cost function evolution of FI-CMA for time varying channel, zero noise and block length $N = 512$ is shown in Figure 6.

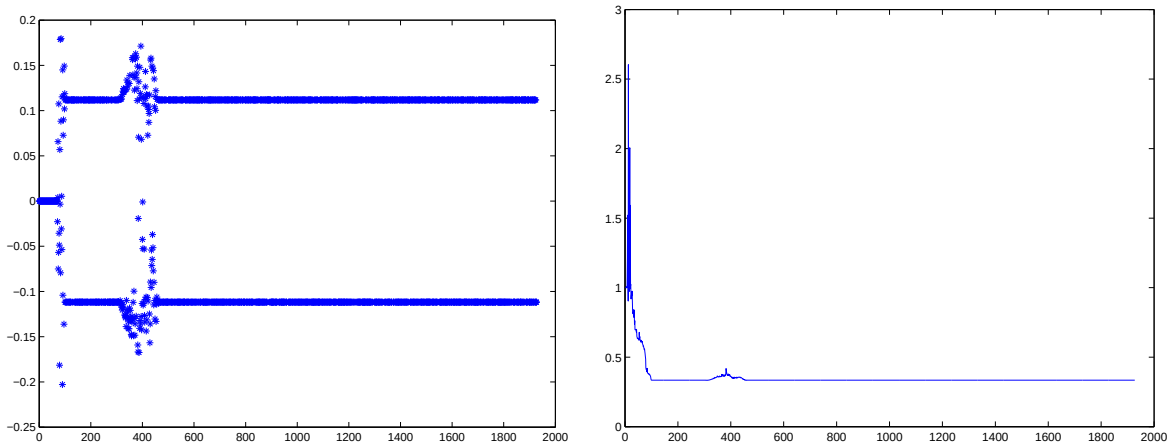

Figure 4: Equalizer output and evolution of cost function **F** for P=2,M=9 with no noise and dynamic **H**

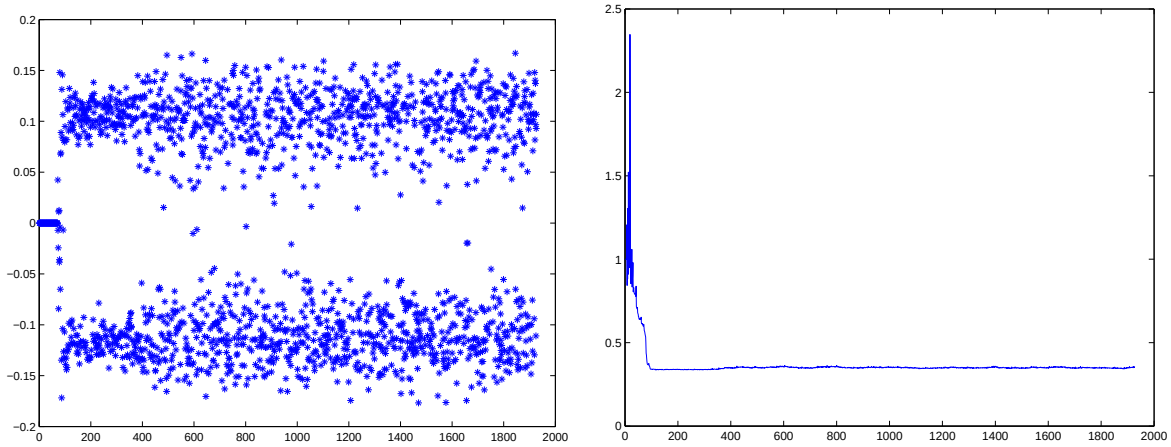

Figure 5: Equalizer output and evolution of cost function **F** for P=2,M=9 with additive Gaussian noise 15db and dynamic **H**

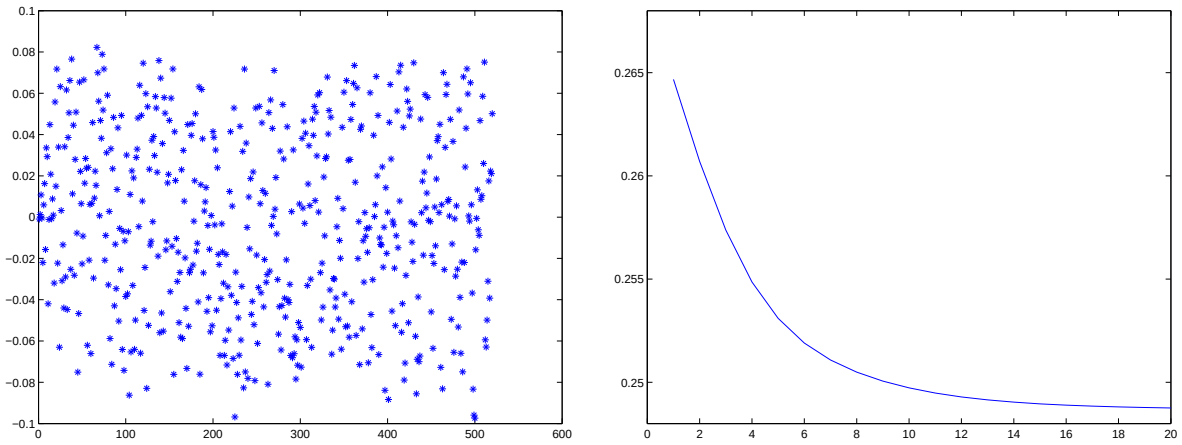

Figure 6: Equalize output and cost function evaluation of the original FI-CMA algorithm for time varying channel for N=512 and no additive noise, 20 iterations

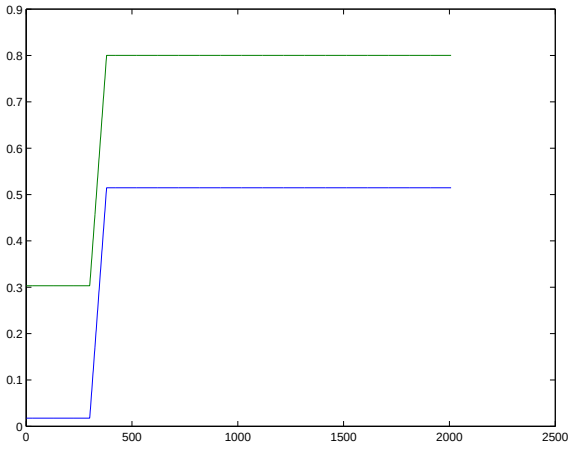

Figure 7: Channel parameter variation

Table 1: computation complexity

Procedure	Complexity	needed for
QR economic [1]	$4(L^2N - L^3/3)$	FI-CMA
QR append row	$3L^2 + 4LK + 8L$	Rec. CMA
QR append column	$3L^2 + 6LK + 8L$	Rec. CMA
QR delete row	$4K^2 + 3L^2 + 6K + 3L$	Rec CMA
QR delete column	0 for the last column	Rec CMA
One FI-CMA itearion	$4NL + 4N + 3L$	both
Rg computation	$(L^2 + L)/2$	Rec CMA

Table 2: Computation complexity evaluated for some N and M

Variable value	R-FI-CMA row update	R-FI-CMA column update	FI-CMA
$N = 2^8$, $M = 7$	5,112,534	8,878,950	27,525,531
$N = 2^8$, $M = 9$	10,736,622	20,212,430	46,296,495
$N = 2^{10}$, $M = 7$	23,546,838	39,185,766	418,260,891
$N = 2^{10}$, $M = 9$	55,795,950	98,218,190	694,486,961

4 Computation complexity

In this section, the computation complexity of different subroutines mentioned in section 2 is compared. Because Recursive and Finite-Interval algorithms do not work with matrics of the same size, we define the following symbols used in Table 1 as:

N - data block size for Finite Interval CMA, stream length for Recursive FI-CMA

M - equalizer order

P - oversampling factor

K - number of rows of **Q** matrix for Recursive FI-CMA, $K = M^2$

L - number of columns of **Q** matrix for FI-CMA and Recursive FI-CMA with column append procedure, size of triangular matrix **R**, $L = P \cdot M$

Since it is very dificult to express cumputation complexity analyticaly, the complexity is numerically evaluated in Table 2 for different N,M with P=2. It is seen that Recursive FI-CMA with row update needs less computation the others.

5 Conclusion and Future work

The propossed algorithm have still fast convergence in comparison to gradient CMA which convergs tipicaly after $\approx 10^4$ iterations. This property is achived thanks to orthonormality of **Q**. In contrast to the FI-CMA proosed Recursive FI-CMA is suitable for deconvolution of the system with time varying parameters and has similar convergency speed and sensibility to additive noise. From Table 2 it is seen that the computation complexity is less or equal to FI-CMA (depending on the parameters). This lower complexity is coused by selecting K as low

[1] my estimates of the complexity of economic QR are higher than presented. Reference were taken from [1]

as possible (i.e. $K = M^2$) which is not optimal block length for FI-CMA as was discussed in section 2.

There ara several things in which the algorithm can be umproved. First of all, the computation of cost function $\mathbf{F}$ in Recursive FI-CMA takes $\mathcal{O}(NM)$ operations and we look for some recursive actualization of $\mathbf{F}$ with less computations. Next, the QR delete-row procedure works with square $\mathbf{Q}$ matrix and the question is if we can use some economic type algorithm, wich work just with first L columns. The above algorithms include many roots, squere-roots and divisions which are very computationaly expensive operations for standard floating point arithmetic. Because of that we plan to implement these algorithms on modern DSPs and compare the floating point arithmetics with Logarithmic Numberigns System (LNS) arithmetic (see [**]). We also plan to parallelize the algorithms and implement them on FPGA with partial dynamic reconfiguration.

References

[1] Charles F. van Loan Gene H. Golub. *Matrix Computations.* Johns Hopkins University Press, 1996.

[2] Phillip A. Regalia. A finite interval constant modulus algorithm. In *Proc. Interrmational Conferene on Acoustics Speech and Signal Processing(ICASSP-2002)*, volume III, pages 2285–2288, Orlando, FL, May 13-17 2002.