Block Algorithms for the Control of Adaptive Antenna Arrays

Zdeněk TOBEŠ
Technical University of Brno
Institute of Radioelectronics
Antonínská 1
662 09 BRNO
Phone: 05 41321234 / 214
Fax: 05 41211135

Abstract

Presented paper deals with the reduction of computational requirements of gradient algorithms for the control of adaptive antenna arrays. Reduction of arithmetical complexity is reached here by the application of a block signal processing to adaptive algorithms. Block versions of the classical Least Mean Square (LMS) algorithm and the Simplified Kalman Filter (SKF) are described in this submission. Adaptation parameters of the presented algorithms are illustrated by results of computer simulations. The block SKF (BSKF) exhibits twice higher computational requirements than LMS, the same misadjustment as LMS and lower rate of convergence than LMS when transversal filters have great number of taps and when relatively high block length of BSKF is used.

1. Introduction

An adaptive antenna array is an antenna system that automatically sets minims of its directivity pattern to directions from those the most powerful interferences come. Desired function of the adaptive antenna is usually reached by the method of the pilot signal [1], [2].

The pilot signal is a signal of exactly given time course that is synchronously transmitted by the transmitter and at the same time generated in the adaptive antenna system. If antenna receives no interferences (only the transmitted pilot signal is received) then the difference between the signal at the antenna output and the generated pilot signal is zero-th after the proper amplification of the output signal of the antenna. On the contrary, if powerful interferences are coming to the antenna array, the difference between the generated pilot signal and the antenna output one (the difference is called the error signal) is non-zeroth and its power is proportional to the power of received interferences. Hence, directivity pattern of the adaptive antenna is synthesized to minimize the mean square of the error signal. If the mean square of the error signal is minimal then the adaptive antenna system "maximally suppresses signals coming from direcfions of the interferences' arrival" or by other words "minims of the antenna pattern are set to the directions of the interferences' arrival".

Synthesis of the antenna pattern is realized by the complex weighting of signals at outputs of antenna elements. Broadband complex weighting is performed by the transversal filters (fig.1). Zero-th tap of the transversal filter is the direct branch of the complex weight the other taps act as the quadrature branches for respective harmonic components of the processed signals. Hence, complex weighting of the broadband signal can be realized by the real weighting of the delay line taps.

Optimal values of complex weights (the Wiener optimum is supposed) are usually sought by gradient algorithms. These algorithms iteratively change setting of weights in contra-direction of the gradient of the mean square error.

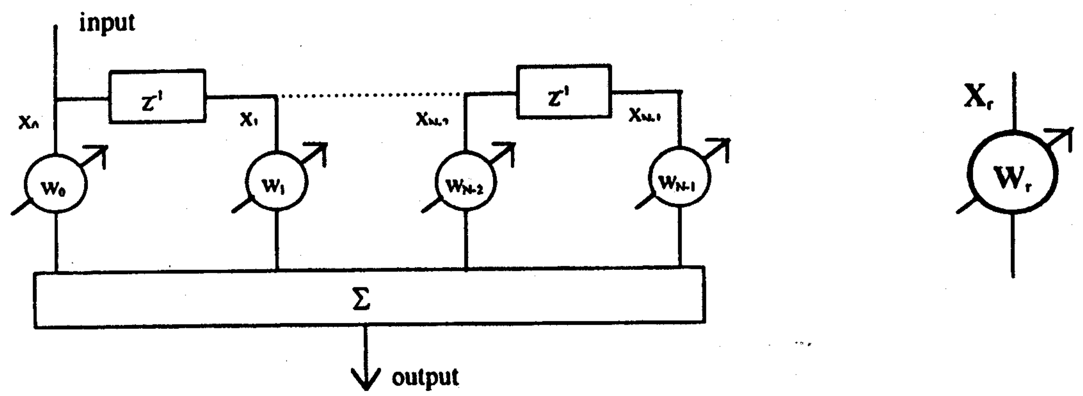

Fig.1 Broadband complex weight realized by the transversal filter

The proper function of the adaptive antenna system is conditioned by sufficiently high number of taps of transversal filters. That is why adaptive setting of filters' weights exhibits high computational requirements in digital systems and why methods of reduction of arithmetical complexity of gradient algorithms are intensively investigated at the present time.

Several ways of the reduction of computational requirements have been so far published [3], [4]. Presented paper reveals new approach to the solving of this problem - application of the block signal processing to adaptive algorithms.

Block version of the Least Mean Square (BLMS) algorithm that has been published in [5] is here applied to the control of adaptive antenna (section II). In section III, block version of the Simplified Kalman filter (BSKF) is derived. Section IV describes results of computer simulations of block algorithms.

In this paper, an antenna array consisting of M omni-directional elements spaced the half wavelength is supposed. Output of each antenna element is completed by the N-tap transversal filter (fig.2).

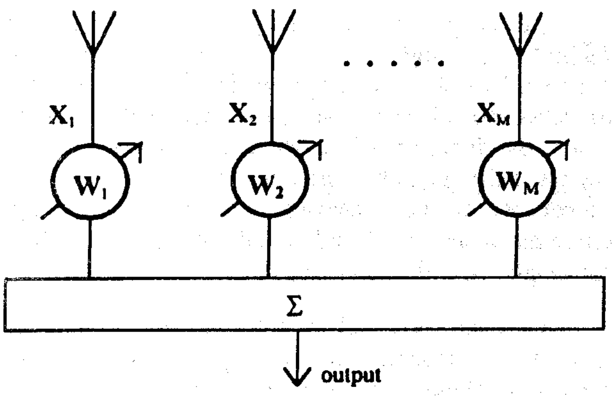

Fig.2 Structure of the antenna array

2. Block version of the LMS algorithm

The LMS is one of the most popular algorithms for the control of adaptive antenna arrays. Relatively low computational requirements and simple both analog and digital implementation are virtues of the LMS, adaptation non stability and relatively poor adaptation parameters are its drawback.

Block version of the LMS algorithm applied to the transversal filter structure has been described in [5]. Derivation of the algorithm [5] modified for the control of adaptive antenna, the structure of which has been described in section I, follows.

Output signal of the antenna array is

$$y(j)=\sum_{r=1}^{M}\sum_{i=0}^{N-1} w_{r,i}(j).x_{r,i}(j)=\sum_{r=1}^{M}\mathbf{W}_r^T(j).\mathbf{X}_r(j)=\sum_{r=1}^{M}\mathbf{X}_r^T(j).\mathbf{W}_r(j) \tag{2.1}$$

where

$\mathbf{W}_r$ is the column vector of optimal weights' estimate of the transversal filter at the output of the r-th antenna element,

$\mathbf{X}_r$ is the tap-input column vector of the transversal filter at the output of the r-th antenna element,

$w_{r,i}$ is the i-th term of $\mathbf{W}_r$ and

$x_{r,i}$ is the i-th term of $\mathbf{X}_r$,

M is number of antenna elements,

N is number of taps of each transversal filter and

T denotes transpose.

Algorithm LMS can be expressed by the following set of equations

$$e(j)=d(j)-\sum_{r=1}^{M}\mathbf{X}_r^T(j)\mathbf{W}_r(j) \tag{2.2a}$$

$$\mathbf{W}_r(j+1)=\mathbf{W}_r(j)+\mu.e(j).\mathbf{X}_r(j) \tag{2.2b}$$

In (2.2), μ is the adaptation constant. μ influences rate of convergence and stability of the algorithm.

Let us write (2.2) at time $j-1$

$$e(j-1)=d(j-1)-\sum_{r=1}^{M}\mathbf{X}_r^T(j-1)\mathbf{W}_r(j-1) \tag{2.3a}$$

$$\mathbf{W}_r(j)=\mathbf{W}_r(j-1)+\mu.e(j-1).\mathbf{X}_r(j-1) \tag{2.3b}$$

Substituting (2.3b) into (2.2a), we obtain

$$e(j)=d(j)-\sum_{r=1}^{M}\mathbf{X}_r^T(j)\mathbf{W}_r(j-1)-\mu e(j-1)\sum_{r=1}^{M}\mathbf{X}_r^T(j)\mathbf{X}_r(j-1)$$

$$=d(j)-\sum_{r=1}^{M}\mathbf{X}_r^T(j).\mathbf{W}_r(j-1)-e(j-1).s(j) \tag{2.4}$$

with

$$s(j)=\mu\sum_{r=1}^{M}\mathbf{X}_r^T(j).\mathbf{X}_r(j-1) \tag{2.5}$$

Combining (2.3a) and (2.4) results in the matrix equation

$$\begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} = \begin{bmatrix} d(j-1) \\ d(j) \end{bmatrix} - \sum_{r=1}^{M} \begin{bmatrix} \mathbf{X}_r^T(j-1) \\ \mathbf{X}_r^T(j) \end{bmatrix} . \mathbf{W}_r(j-1) - \begin{bmatrix} 0 & 0 \\ s(j) & 0 \end{bmatrix} . \begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} . \tag{2.6}$$

Rearranging (2.6) yields

$$\begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(j) & 1 \end{bmatrix} \left\{ \begin{bmatrix} d(j-1) \\ d(j) \end{bmatrix} - \sum_{r=1}^{M} \begin{bmatrix} \mathbf{X}_r^T(j-1) \\ \mathbf{X}_r^T(j) \end{bmatrix} \mathbf{W}_r(j-1) \right\} . \tag{2.7}$$

Now, let us turn our attention to the last term of (2.7) - even- and odd-numbered terms of the involved vectors will be grouped

$$\begin{bmatrix} \mathbf{X}_r^T(j-1) \\ \mathbf{X}_r^T(j) \end{bmatrix} . \mathbf{W}_r(j-1) = \begin{bmatrix} x_{r,0}(j-1) & x_{r,1}(j-1) & \dots & x_{r,N-1}(j-1) \\ x_{r,0}(j) & x_{r,1}(j) & \dots & x_{r,N-1}(j) \end{bmatrix} . \begin{bmatrix} w_{r,0}(j-1) \\ w_{r,1}(j-1) \\ \vdots \\ w_{r,N-1}(j-1) \end{bmatrix} =$$

$$= \begin{bmatrix} x_r(j-1) & x_r(j-2) & \dots & x_r(j-N) \\ x_r(j) & x_r(j-1) & \dots & x_r(j-N+1) \end{bmatrix} . \begin{bmatrix} w_{r,0}(j-1) \\ w_{r,1}(j-1) \\ \vdots \\ w_{r,N-1}(j-1) \end{bmatrix} =$$

$$= \begin{bmatrix} x_r(j-1) & x_r(j-3) & \dots & x_r(j-N+1) & x_r(j-2) & x_r(j-4) & \dots & x_r(j-N) \\ x_r(j) & x_r(j-2) & \dots & x_r(j-N+2) & x_r(j-1) & x_r(j-3) & \dots & x_r(j-N+1) \end{bmatrix} . \begin{bmatrix} w_{r,0}(j-1) \\ w_{r,2}(j-1) \\ \vdots \\ w_{r,N-2}(j-1) \\ w_{r,1}(j-1) \\ w_{r,3}(j-1) \\ \vdots \\ w_{r,N-1}(j-1) \end{bmatrix} \tag{2.8}$$

Let us assume that transversal filters have even number of taps N. Then (2.8) can be expressed in more compact form

$$\begin{bmatrix} \mathbf{X}_r^T(j-1) \\ \mathbf{X}_r^T(j) \end{bmatrix} . \mathbf{W}_r(j-1) = \begin{bmatrix} \mathbf{A}_{r,1}(j) & \mathbf{A}_{r,2}(j) \\ \mathbf{A}_{r,0}(j) & \mathbf{A}_{r,1}(j) \end{bmatrix} \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} \tag{2.9}$$

where

$$\mathbf{A}_{r,0}(j) = \begin{bmatrix} x_r(j) & x_r(j-2) & \dots & x_r(j-N+2) \end{bmatrix}$$

$$\mathbf{A}_{r,1}(j) = \begin{bmatrix} x_r(j-1) & x_r(j-3) & \dots & x_r(j-N+1) \end{bmatrix}$$

$$\mathbf{A}_{r,2}(j) = \begin{bmatrix} x_r(j-2) & x_r(j-4) & \dots & x_r(j-N) \end{bmatrix}$$

$$\mathbf{W}_{r,0}(j-1) = \begin{bmatrix} w_{r,0}(j-1) & w_{r,2}(j-1) & \dots & w_{r,N-2}(j-1) \end{bmatrix}^T$$

$$\mathbf{W}_{r,1}(j-1) = \begin{bmatrix} w_{r,1}(j-1) & w_{r,3}(j-1) & \dots & w_{r,N-1}(j-1) \end{bmatrix}^T \tag{2.10}$$

Substituting (2.9) to (2.7) yields

$$\begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(j) & 1 \end{bmatrix} \left\{ \begin{bmatrix} d(j-1) \\ d(j) \end{bmatrix} - \sum_{r=1}^{M} \begin{bmatrix} \mathbf{A}_{r,1}(j) & \mathbf{A}_{r,2}(j) \\ \mathbf{A}_{r,0}(j) & \mathbf{A}_{r,1}(j) \end{bmatrix} \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} \right\} \tag{2.11}$$

The same kind of work that has been performed for the computing of error signals can be done for updating weights. First substitute (2.3b) into (2.2b)

$$\mathbf{W}_r(j+1)=\mathbf{W}_r(j-1)+\mu.e(j).\mathbf{X}_r(j)+\mu.e(j-1).\mathbf{X}_r(j-1) \tag{2.12}$$

or, with the above notations

$$\begin{bmatrix}\mathbf{W}_{r,0}(j+1)\\ \mathbf{W}_{r,1}(j+1)\end{bmatrix}=\begin{bmatrix}\mathbf{W}_{r,0}(j-1)\\ \mathbf{W}_{r,1}(j-1)\end{bmatrix}+\mu.e(j).\begin{bmatrix}\mathbf{A}^T_{r,0}(j)\\ \mathbf{A}^T_{r,1}(j)\end{bmatrix}+\mu.e(j-1).\begin{bmatrix}\mathbf{A}^T_{r,1}(j)\\ \mathbf{A}^T_{r,2}(j)\end{bmatrix}=$$

$$=\begin{bmatrix}\mathbf{W}_{r,0}(j-1)\\ \mathbf{W}_{r,1}(j-1)\end{bmatrix}+\mu.\begin{bmatrix}\mathbf{A}^T_{r,1}(j) & \mathbf{A}^T_{r,0}(j)\\ \mathbf{A}^T_{r,2}(j) & \mathbf{A}^T_{r,1}(j)\end{bmatrix}.\begin{bmatrix}e(j-1)\\ e(j)\end{bmatrix}. \tag{2.13}$$

Equations (2.11) and (2.13)are the exact equivalent of the classical LMS algorithm (2.2). Reduction in arithmetic complexity can take place by rewriting (2.11) and (2.13) as

$$\begin{bmatrix}e(j-1)\\ e(j)\end{bmatrix}=\begin{bmatrix}1 & 0\\ -s(j) & 1\end{bmatrix}.\left\{\begin{bmatrix}d(j-1)\\ d(j)\end{bmatrix}-\sum_{r=1}^{M}\begin{bmatrix}\mathbf{A}_{r,1}(j).[\mathbf{W}_{r,0}(j-1)+\mathbf{W}_{r,1}(j-1)]+[\mathbf{A}_{r,2}(j)-\mathbf{A}_{r,1}(j)].\mathbf{W}_{r,1}(j-1)\\ \mathbf{A}_{r,1}(j).[\mathbf{W}_{r,0}(j-1)+\mathbf{W}_{r,1}(j-1)]-[\mathbf{A}_{r,1}(j)-\mathbf{A}_{r,0}(j)].\mathbf{W}_{r,0}(j-1)\end{bmatrix}\right\}$$

$$\begin{bmatrix}\mathbf{W}_{r,0}(j+1)\\ \mathbf{W}_{r,1}(j+1)\end{bmatrix}=\begin{bmatrix}\mathbf{W}_{r,0}(j-1)\\ \mathbf{W}_{r,1}(j-1)\end{bmatrix}+\mu\begin{bmatrix}\mathbf{A}^T_{r,1}(j).[e(j-1)+e(j)]-[\mathbf{A}_{r,1}(j)-\mathbf{A}_{r,0}(j)]^T.e(j)\\ \mathbf{A}^T_{r,1}(j).[e(j-1)+e(j)]+[\mathbf{A}_{r,2}(j)-\mathbf{A}_{r,1}(j)]^T.e(j-1)\end{bmatrix}.$$

The filtering operation

$$\mathbf{A}_{r,1}(j).[\mathbf{W}_{r,0}(j-1)+\mathbf{W}_{r,1}(j-1)]$$

is common between the two terms of (2.14a), and the overall arithmetic computation is now that of three length $N/2$ filters, instead of four. Computations in (2.14b) are similar .

The term $s(j)$,that has been defined as a scalar product of length N, can be computed recursively from $s(j-2)$ as

$$s(j)=s(j-2)+\mu.\sum_{r=1}^{M}\{x_r(j-1).[x_r(j)+x_r(j-2)]-$$

$$-x_r(j-N-1).[x_r(j-N)+x_r(j-N-2)]\} \tag{2.15}$$

This algorithm is provided for block size $L=2$, which means that error signal and weights are computed once during two samples. Arbitrary block size is defined similary. Larger block size provided larger reduction of aritmetical complexity. Generalization of the algorithm (2.14) to an arbitrary block size is described in [9].

3. Block version of the Simplified Kalman Filter

Simplified Kalman filter (SKF) and the analysis of its parameters have been described in [6], [7] and [8]. SKF excels in very good adaptation and numerical stability, relatively high rate of convergence and relatively low computational requirements. SKF can be expressed by the following set of equations

$$\hat{w}_{r,i}(j+1)=\hat{w}_{r,i}(j)+k_{r,i}(j).e(j) \tag{3.1a}$$

$$k_{r,i}(j)=\frac{p_{r,i}(j).x_{r,i}(j)}{\sum_{r=1}^{M}\sum_{i=0}^{N-1}p_{r,i}(j).x^2_{r,i}(j)+R} \tag{3.1b}$$

$$p_{r,i}(j+1)=p_{r,i}(j).[1-k_{r,i}(j).x_{r,i}(j)]. \tag{3.1c}$$

where

$\hat{w}_{r,i}$ is estimate of i-th optimal weight of the r-th transversal filter,

$k_{r,i}$ denotes i-th component of the Kalman gain vector of the r-th transversal filter,

e is the error signal (difference between the pilot signal $d(j)$ and the output signal of the antenna)

$$e(j) = d(j) - \sum_{r=1}^{M} \mathbf{X}_r^T(j).\mathbf{W}_r(j) \quad , \tag{3.1d}$$

($\mathbf{W}_r$ is the column vector of $\hat{w}_{r,i}$ and $\mathbf{X}_r$ is the column vector of $x_{r,i}$),

$x_{r,i}$ is the sample at the i-th tap of the r-th filter,

$p_{r,i}$ is the variance of the estimation error of i-th optimal weight at the r-th filter and

R is the variance of residual error.

Equation (3.1a) can be rewritten to the matrix form

$$\mathbf{W}_r(j+1) = \mathbf{W}_r(j) + e(j).\mathbf{K}_r(j)$$

($\mathbf{K}_r$ is the column vector of $k_{r,i}$).

Thanks to the identity of (2.2a) and (3.1d) and thanks to the similarity between (2.2b) and (3.1a), an application of the block signal processing, described in II, to SKF seems to be possible.

As previously, (3.1d) and (3.1a) are expressed at time $j-1$

$$\mathbf{W}_r(j) = \mathbf{W}_r(j-1) + e(j-1).\mathbf{K}_r(j-1) \tag{3.2a}$$

$$e(j-1) = d(j-1) - \sum_{r=1}^{M} \mathbf{X}_r^T(j-1).\mathbf{W}_r(j-1) \quad . \tag{3.2d}$$

Substituting (3.2a) into (3.1d), we obtain

$$e(j) = d(j) - \sum_{r=1}^{M} \mathbf{X}_r^T(j).\mathbf{W}_r(j-1) - e(j-1)\sum_{r=1}^{M} \mathbf{X}_r^T(j).\mathbf{K}_r(j-1) =$$

$$= d(j) - \sum_{r=1}^{M} \mathbf{X}_r^T(j).\mathbf{W}_r(j-1) - e(j-1).s(j) \tag{3.3}$$

with $s(j) = \sum_{r=1}^{M} \mathbf{X}_r^T(j).\mathbf{K}_r(j-1)$. (3.4)

Since (3.3) is totally identical with (2.4), (3.3) can be directly rewritten to the form of (2.10)

$$\begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(j) & 1 \end{bmatrix} \left\{ \begin{bmatrix} d(j-1) \\ d(j) \end{bmatrix} - \sum_{r=1}^{M} \begin{bmatrix} \mathbf{A}_{r,1}(j) & \mathbf{A}_{r,2}(j) \\ \mathbf{A}_{r,0}(j) & \mathbf{A}_{r,1}(j) \end{bmatrix} \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} \right\} \tag{3.5}$$

where vectors $\mathbf{A}$ and $\mathbf{W}$ are given by (2.10).

Now, let us turn our attention to (3.1a). Substituting (3.2a) into (3.1a) yields

$$\mathbf{W}_r(j+1) = \mathbf{W}_r(j-1) + e(j).\mathbf{K}_r(j) + e(j-1).\mathbf{K}_r(j-1) \tag{3.6}$$

Time dependence of the Kalman gain vector disables application of the blocking procedure used in the algorithm LMS. That is why the constant value of Kalman gain is supposed during the block duration. This assumption reduces computational requirements of SKF from the "non-block" reason. Unfortunately, the described simplification negatively influences adaptation parameters of the algorithm (see computer simulations).

Components of the Kalman gain are shifted by the same way as input samples in delay lines. With this idea in mind, let us define

$$\mathbf{K}_{r,0}(j) = \left[k_{r,0}(j) \quad k_{r,2}(j) \quad \ldots \quad k_{r,N-2}(j) \right]$$

$$\mathbf{K}_{r,1}(j) = \left[k_{r,1}(j) \quad k_{r,3}(j) \quad \ldots \quad x_{r,N-1}(j) \right] \tag{3.7}$$

$$\mathbf{K}_{r,2}(j) = \left[k_{r,2}(j) \quad k_{r,4}(j) \quad \ldots \quad k_{r,N}(j) \right] \; .$$

Equation (3.7) contains $N+1$ terms of the Kalman gain vector. The "new" term $k_{r,N}(j)$ can be computed e.g. according to the relation

$$k_{r,N}(j) = \frac{p_{r,N}(j).x_{r,N}(j)}{\sum_{r=1}^{M}\sum_{i=1}^{N} p_{r,i}(j).x_{r,i}^2(j) + R} \; .$$

Second new term $p_{r,N}$ is necessary for computing $k_{r,N}$. This term is computed according relation (3.1c).For more details see [9].

Equation (3.6) can be rewritten on the base of (3.7) to the form

$$\begin{bmatrix} \mathbf{W}_{r,0}(j+1) \\ \mathbf{W}_{r,1}(j+1) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} + e(j).\begin{bmatrix} \mathbf{K}_{r,0}^T(j) \\ \mathbf{K}_{r,1}^T(j) \end{bmatrix} + e(j-1).\begin{bmatrix} \mathbf{K}_{r,1}^T(j) \\ \mathbf{K}_{r,2}^T(j) \end{bmatrix} =$$

$$= \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{r,1}^T(j) & \mathbf{K}_{r,0}^T(j) \\ \mathbf{K}_{r,2}^T(j) & \mathbf{K}_{r,1}^T(j) \end{bmatrix} \begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} \tag{3.8}$$

Rearranging (3.5) and (3.8)

$$\begin{bmatrix} e(j-1) \\ e(j) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -s(j) & 1 \end{bmatrix} \cdot \left\{ \begin{bmatrix} d(j-1) \\ d(j) \end{bmatrix} - \sum_{r=1}^{M} \begin{bmatrix} \mathbf{A}_{r,1}(j).\left[\mathbf{W}_{r,0}(j-1)+\mathbf{W}_{r,1}(j-1)\right] + \left[\mathbf{A}_{r,2}(j)-\mathbf{A}_{r,1}(j)\right].\mathbf{W}_{r,1}(j-1) \\ \mathbf{A}_{r,1}(j).\left[\mathbf{W}_{r,0}(j-1)+\mathbf{W}_{r,1}(j-1)\right] - \left[\mathbf{A}_{r,1}(j)-\mathbf{A}_{r,0}(j)\right].\mathbf{W}_{r,0}(j-1) \end{bmatrix} \right\}$$

$$\begin{bmatrix} \mathbf{W}_{r,0}(j+1) \\ \mathbf{W}_{r,1}(j+1) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{r,0}(j-1) \\ \mathbf{W}_{r,1}(j-1) \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{r,1}^{T}(j).\left[e(j-1)+e(j)\right] - \left[\mathbf{K}_{r,1}(j)-\mathbf{K}_{r,0}(j)\right]^{T}.e(j) \\ \mathbf{K}_{r,1}^{T}(j).\left[e(j-1)+e(j)\right] + \left[\mathbf{K}_{r,2}(j)-\mathbf{K}_{r,1}(j)\right]^{T}.e(j-1) \end{bmatrix} \qquad (3.9a,b)$$

leads to the additional reduction of arithmetic complexity.

The term $s(j)$ given by (3.4) is computed according to the relation

$$s(j) = s(j-2) + \sum_{r=1}^{M} \left\{ k_{r,1}(j).x_r(j) + k_{r,2}(j).x_r(j-1) - k_{r,1}(j-N).x_r(j-N) - k_{r,2}(j-N).x_r(j-N-1) \right) .$$

Generalization of the algorithm (3.9) to an arbitrary block size is described in [9].

4. Computer simulations

Developed block algorithms have been tested by computer simulations of the adaptive antenna array consisting of three half-wavelength-spaced omni-directional elements. Desired signal (white, variance 0.01) has come perpendicularly to the antenna aperture, direction of the interference signal (white, variance 1) has declined 45 degrees from the direction of the desired signal.

At the first time, computational requirements of the control algorithms have been measured by Turbo Profiler (version 2.2, Borland International). The block length of block algorithms has been set to L=2 because algorithms of this minimal block length exhibit the maximal arithmetical complexity. Time required by processor for performing the classical LMS algorithm is taken as 100% (tab.1). Computational requirements of SKF are approximately 3-5 times higher. For low number of taps, computational requirements of BLMS are higher than those of classical LMS. If number of taps is increased then arithmetical complexity of BLMS significantly drops. This effect is caused by the necessity of computing certain help variables in BLMS - if the transversal filter consists of high number of taps then these help variables are used many times and relative time for their computing drops. Computational requirements of BSKF are all the time lower than SKF exhibits. For high number of taps, arithmetical complexity of BSKF is twice higher in comparison with the classical LMS.

Later, convergence properties of the control algorithms have been tested. Adaptation constant of the LMS and the BLMS algorithms has been set to μ=3.10^{-3}, the variance of residual error R=10^{-6} and all the variances of estimation errors $p_{r,i}$=0.1 (r=1,2..M, i=0,1..N-1) have been used for the SKF and the BSKF algorithms in the following simulations.

Figure 3 shows the time courses of ensembly averaged (20 ensembles) square error of LMS (dotted) and SKF (solid) algorithms. Transversal filters have consisted of 10 taps. Since there are no significant differences between time courses of LMS and SKF, block algorithms are compared only with the LMS in further.

Figure 4 shows time courses of ensembly averaged (20 ensembles) square error of LMS (dashed), BLMS (dotted) and BSKF (solid). Transversal filters have again consisted of 10 taps. The block length has been L=2. Learning curves of LMS and BLMS algorithms are approximately the same. BSKF exhibits lower rate of convergence and higher misadjustment than LMS family.

Table 1 Dependence of computational requirements of thecontrol algorithms on the number of taps of transversal filters. Time required by the processor for performing the classical LMS algorithm is taken as 100%.

taps	LMS	SKF	Block LMS	Block SKF
3 x 4	100	503	224	406
3 x 16	100	299	98	228
3 x 32	100	290	88	228

Figure 5 shows time courses of ensembly averaged (20 ensembles) square error of LMS (dashed), BLMS (dotted) and BSKF (solid). Transversal filters have consisted of 30 taps. The block length has been $L=10$. At this situation, the learning curves of BLMS and BSKF algorithms exhibit lower rate of convergence than the classical LMS. Misadjustment of BSKF in on the level of LMS algorithm whereas the misadjustment of BLMS is higher.

5. Conclusion

Presented paper describes block algorithms for the control of adaptive antennas based on the pilot signal method.

Block Least Mean Square (BLMS) algorithm is derived from the popular LMS algorithm. Computational requirements of BLMS drops when number of taps of transversal filters raises and when the block length grows. If the block length is short then adaptation parameters of BLMS are the same as the classical LMS exhibits. Increase of the block length causes the growth of misadjustment and reduction of the convergence rate of BLMS.

Block Simplified Kalman filter (BSKF) is based on the SKF algorithm. Computational requirements of BSKF decreases when block length grows and are relatively independent on the number of taps. Requirements of BSKF are approximately twice higher than the classical LMS exhibits. If block length raises then misadjusment of BSKF converges to the misadjustment of LMS and convergence rate of BSKF grows.

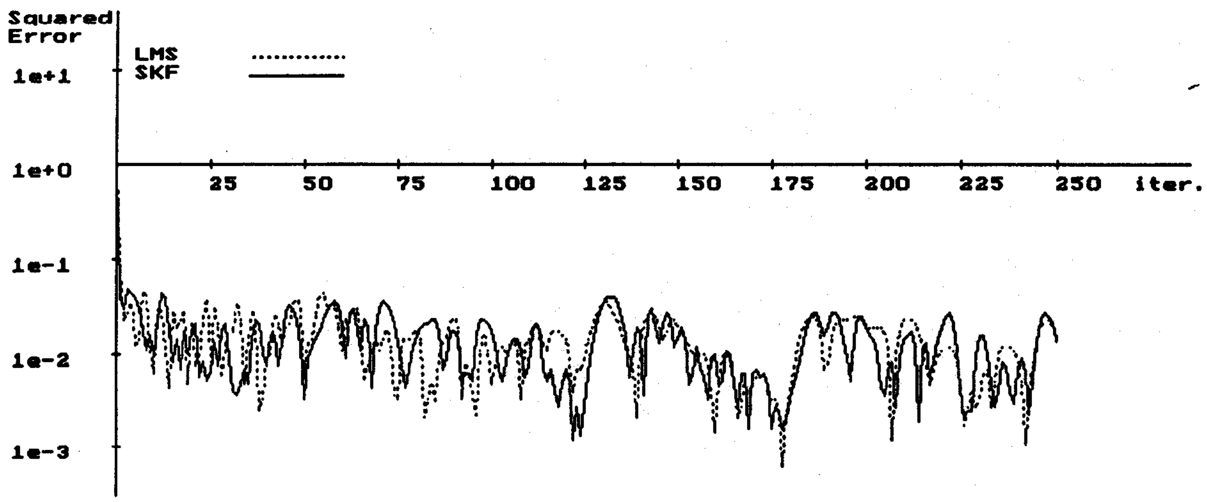

Fig.3 Time courses of the LMS (dotted) and SKF (solid) algorithms. Transversal filters consist of 10 taps.

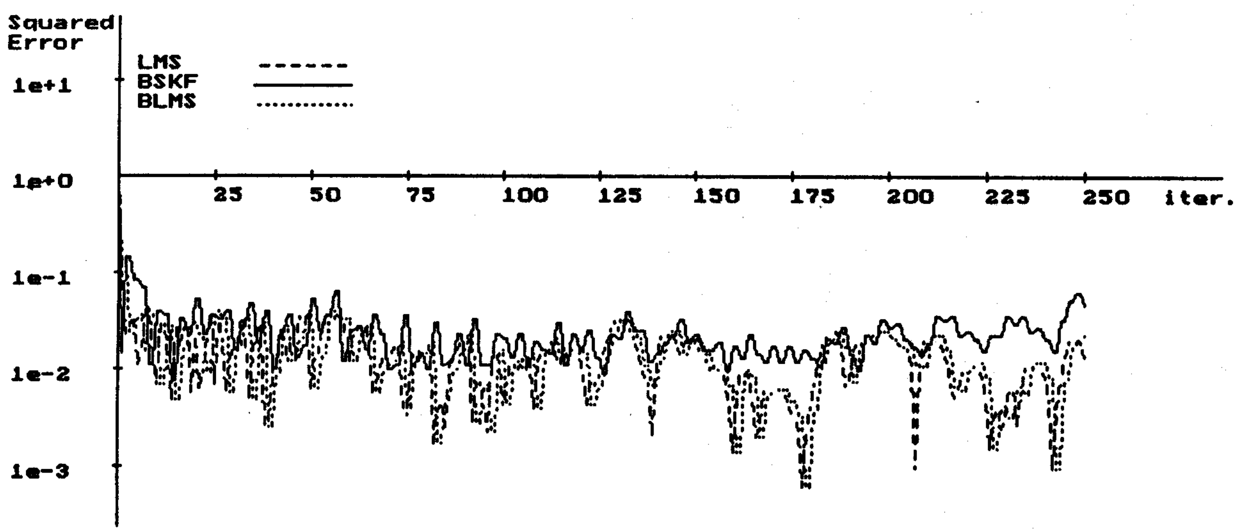

Fig.4 Time courses of the LMS (dashed) BLMS (dotted) and BSKF (solid) algorithms. Block length L=2. Transversal filters consist of 10 taps.

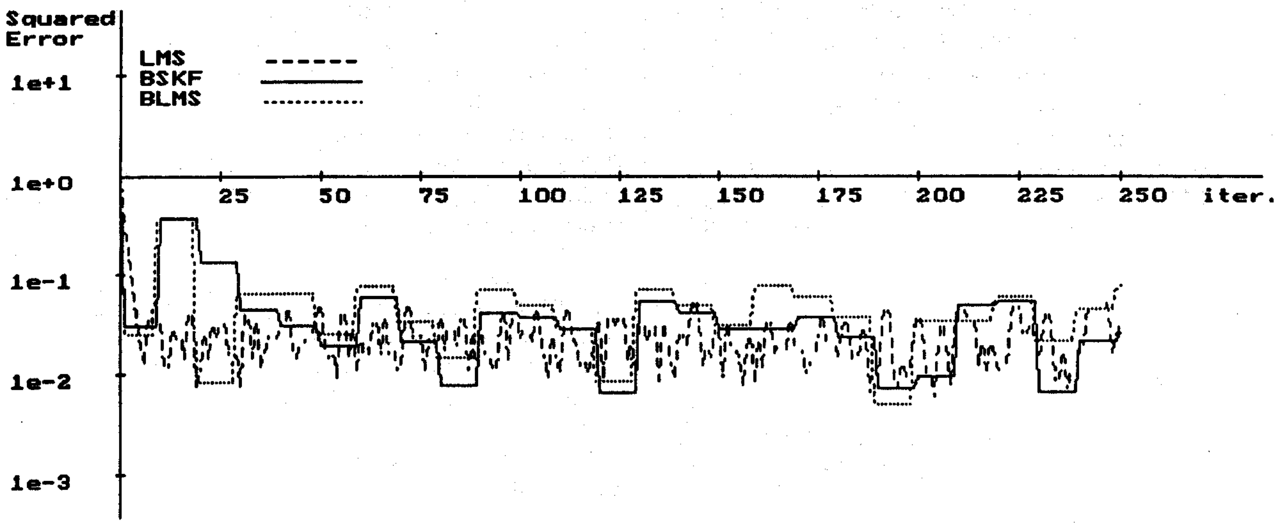

Fig.5 Time courses of the LMS (dashed) BLMS (dotted) and BSKF (solid) algorithms. Block length L=10. Transversal filters consist of 30 taps.

The BSKF algorithm is both numerically and adaptively very stable and is independent on a priori information about signals' statistics [6], [8]. On the contrary, the BLMS algorithm requires the knowledge of the trace of autocorrelation matrix of the tap-input vector for the proper setting of the adaptation constant μ [5], [8].

Taking all the above facts in mind, we can conclude that the BSKF algorithm is suitable for the digital control of adaptive antenna arrays when transversal filters have great number of taps and when relatively high block length of BSKF is used. Then the BSKF has twice higher computational requirements than LMS, the same misadjustment as LMS and lower rate of convergence than LMS.

References

[1] Widrow,B.- Mantey,P.E.- Griffiths,L.J.- Goode,B.B.: Adaptive Antenna Systems. Proc.IEEE, vol.55, December 1967, No.12, pp.2143-2159.

[2] Widrow,B.- McCool,J.M.: Comparison of Adaptive Algorithms Based on the Method of Steepest Descent and Random Search. Proc.IEEE, vol. AP-24, September 1976, No.5, pp.615-637.

[3] Cantoni,A.: Application of Orthogonal Perturbation Sequences to Adaptive Beamforming. Proc.IEEE, vol. AP-28, March 1980, No.3, pp.191-202.

[4] Chapman,J.D.: Partial Adaptivity for the Large Array. Proc.IEEE, vol. AP-24, September 1976, No.5, pp.685-696.

[5] Benesty,J.- Duhamel,P.: A Fast Exact Least Mean Square Adaptive Algorithm. Proc.IEEE, vol. SP-40, December 1992, No.12, pp.2904-2919.

[6] Raida,Z.: Steering an Adaptive Antenna Array by the Simplified Kalman Filter. Proc.IEEE, vol. AP-43, June 1995, No.6, pp.

[7] Raida,Z.: Digitally Implementable Steering Algorithms Based on Kalman Filter. Proc. of Int. symposium on antennas JINA'94, Nice, France, pp.618-621.

[8] Raida,Z.: Comparison of Adaptive Antenna Arrays Controlled by Gradient Algorithms. Radioengineering, vol.3, September 1994, No.3, pp.6-14.

[9] Tobeš,Z.: The broadband adaptive antenna. Diploma work, Faculty of Electrical Engineering and Computer Science, Technical University Brno, 1995.

About author,...

Zdeněk TOBEŠ was born in Hranice na Moravě, Czechoslovakia, in 1971. He received the MSc. degree in electrical engineering from VUT Brno, in 1995. At present, he is the PhD student at VUT Brno, Institute of Radioelectronics. He is interested in adaptive antenna systems .