A RECURSIVE ALGORITHM FOR ADAPTIVE ARRAYS

Yoshinori MIKI and Kazuaki TAKAO
Dept. of Electrical Engineering, Kyoto University, Kyoto 606, Japan

1 Introduction

Adaptive arrays under directionally constrained minimization of power (DCMP) method perform to minimize the array output power while maintaining a constant response to the desired signal at the specified direction and frequency [1].

As to the transient performance, the steepest gradient algorithm does not converge fast when eigenvalues of the input correlation matrix are diverged widely. On the other hand, the sampled matrix inversion (SMI) algorithm does not suffer such effect from the adverse input environment. It, however, is vulnerable to hardware errors such as weight values since the output is not fed back to the control.

In this paper we consider the recursive algorithm [2], which takes care of this susceptibility of SMI. Examples will be shown to demonstrate the improvement via computer simulation.

2 Principle

We consider the case that incident waves are all narrow-banded, and denote the inputs and variable weights by complex expressions in the form of column vectors, $\boldsymbol{X}$ and $\boldsymbol{W}$, respectively. Thus, the output voltage, y, and power, p_{out}, of the array can be expressed as follows.

$$y = \boldsymbol{W}^{\text{H}}\boldsymbol{X} = \boldsymbol{X}^{\text{t}}\boldsymbol{W}^{*}, \quad p_{\text{out}} = yy^{*} = \boldsymbol{W}^{\text{H}}R_{\text{xx}}\boldsymbol{W}, \quad R_{\text{xx}} = \text{E}[\boldsymbol{X}\boldsymbol{X}^{\text{H}}] \tag{1}$$

where the superscripts $^{\text{t}}$, * and $^{\text{H}}$ denote transpose, conjugate and conjugate transpose (Hermitian), respectively. R_{xx} is the correlation matrix of input, and $\text{E}[\cdot]$ denotes expectation.

The principle of DCMP can be expressed as follows,

$$\text{minimize} \quad \boldsymbol{W}^{\text{H}}R_{\text{xx}}\boldsymbol{W} \quad \text{for } \boldsymbol{W}, \qquad \text{subject to} \quad \boldsymbol{C}^{\text{t}}\boldsymbol{W}^{*} = \boldsymbol{H} \tag{2}$$

where $\boldsymbol{C}$ and $\boldsymbol{H}$ are the constraint matrix and constraint response vector with respect to the specified direction, respectively. The optimum weight vector is given, as a closed form, by

$$\boldsymbol{W}_{\text{opt}} = R_{\text{xx}}^{-1}[\boldsymbol{C}^{\text{H}}R_{\text{xx}}^{-1}\boldsymbol{C}]^{-1}\boldsymbol{H}^{*} \tag{3}$$

Sampled matrix method estimates the R_{xx} to be used in (3) by sampled input as in the following:

$$R_{\text{xx}}(k+1) = \beta R_{\text{xx}}(k) + (1-\beta)\boldsymbol{X}(k)\boldsymbol{X}^{\text{H}}(k) \tag{4}$$

where β is "forgetting factor," satisfying $0 < \beta < 1$. A matrix inversion formula is applied to (4), and SMI uses the following equations.

$$R_{\rm xx}^{-1}(k+1) = \frac{1}{\beta}R_{\rm xx}^{-1}(k) + \frac{1-\beta}{\beta} \cdot \frac{R_{\rm xx}^{-1}(k)\boldsymbol{X}(k)\boldsymbol{X}^{\rm H}(k)R_{\rm xx}^{-1}(k)}{\beta - (1-\beta)\boldsymbol{X}^{\rm H}(k)R_{\rm xx}^{-1}(k)\boldsymbol{X}(k)} \tag{5}$$

$$\boldsymbol{W}(k+1) = R_{\rm xx}^{-1}(k+1)[\boldsymbol{C}^{\rm H}R_{\rm xx}^{-1}(k+1)\boldsymbol{C}]^{-1}\boldsymbol{H}^* \tag{6}$$

On the other hand, we can separate $\boldsymbol{W}$ into constant weight $\boldsymbol{F}$ which forms the mainbeam to the desired direction, and variable weight $\boldsymbol{V}$ which is used to minimize the interference. Thus, we have $\boldsymbol{W} = \boldsymbol{F} + P\boldsymbol{V}$, where P is the projection matrix which orthogonalizes V to the desired signal vector. Eq. (2) can be rewritten as

$$\text{minimize} \quad p_{\rm out} = (\boldsymbol{F} + P\boldsymbol{V})^{\rm H}R_{\rm xx}(\boldsymbol{F} + P\boldsymbol{V}) \quad \text{for } \boldsymbol{V} \tag{7}$$

and the optimum values of $\boldsymbol{V}$ and $\boldsymbol{W}$ are obtained by the closed forms [2],

$$\boldsymbol{V}_{\rm opt} = -QPR_{\rm xx}\boldsymbol{F}, \quad \boldsymbol{W}_{\rm opt} = \boldsymbol{F} - P\boldsymbol{V}_{\rm opt} \tag{8}$$

where $Q \equiv (PR_{\rm xx}P)^+$ and the superscript $^+$ denotes pseudo inverse. Applying (4) to (8), the following algorithms are derived [2].

$$Q(k+1) = \frac{1}{\beta}Q(k) + \frac{1-\beta}{\beta} \cdot \frac{Q(k)P\boldsymbol{X}(k)\boldsymbol{X}^{\rm H}(k)PQ(k)}{\beta + (1-\beta)\boldsymbol{X}^{\rm H}(k)Q(k)\boldsymbol{X}(k)} \tag{9}$$

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - (1-\beta)Q(k+1)P\boldsymbol{X}(k)y^*(k) \tag{10}$$

It should be noted that (10) involves the output y, forming a feedback loop. Thus, both the rapid convergence and immunity to hardware errors are established in this algorithm.

3 Numerical examples

In order to compare transient performances of the SMI and recursive algorithms, computer simulations were carried out on 4-element, equispaced linear array with the spacing of a half wavelength. In one iteration, ten samples were used for smoothing. We assume that, as shown in Table 1, two monochromatic plane waves, the desired signal and interference, are incident on the array. Thermal noise is assumed to be of the same magnitude and independent at each input. The actual weight $\widetilde{\boldsymbol{W}}$ has constant error, $\Delta\boldsymbol{W}$, *i.e.*

$$\widetilde{\boldsymbol{W}} = \boldsymbol{W} + \Delta\boldsymbol{W} \tag{11}$$

where $\boldsymbol{W}$ is the nominal weight. Each element of the vector $\Delta\boldsymbol{W}$ assumed to be independent and follow Gaussian distribution.

Fig.1 shows the learning curves of the desired signal and interference components in the outputs of the SMI and recursive algorithms. In this case, the relative error is 1% (the standard deviation). Fig.2 shows the results for the case where the error is 5%.

In both figures, it is evident that the proposed recursive algorithm is able to compensate for the weight errors. This can be understood as follows.

Considering that $y^* = \boldsymbol{X}^{\mathrm{H}}\widetilde{\boldsymbol{W}}$, we rewrite (10) by

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) - (1-\beta)Q(k+1)P\boldsymbol{X}(k)\boldsymbol{X}^{\mathrm{H}}(k)\widetilde{\boldsymbol{W}}(k) \tag{12}$$

With (12) , we obtain

$$\widetilde{\boldsymbol{W}}(k+1) = \widetilde{\boldsymbol{W}}(k) - (1-\beta)Q(k+1)P\boldsymbol{X}(k)\boldsymbol{X}^{\mathrm{H}}(k)\widetilde{\boldsymbol{W}}(k) \tag{13}$$

which means that the actual weight $\widetilde{\boldsymbol{W}}$ is updated *rightly* by (10).

4 Conclusion

The merits of the recursive algorithm for adaptive arrays are shown via computer simulation. The consideration to reduce the computing time per iteration and also to apply the concept to hardware errors in general is under study.

Acknowledgement

The authors wish to thank Prof. I. Kimura and his group for their helpful discussions. It must be added that all computer simulations are carried out on Data Processing Center of Kyoto University.

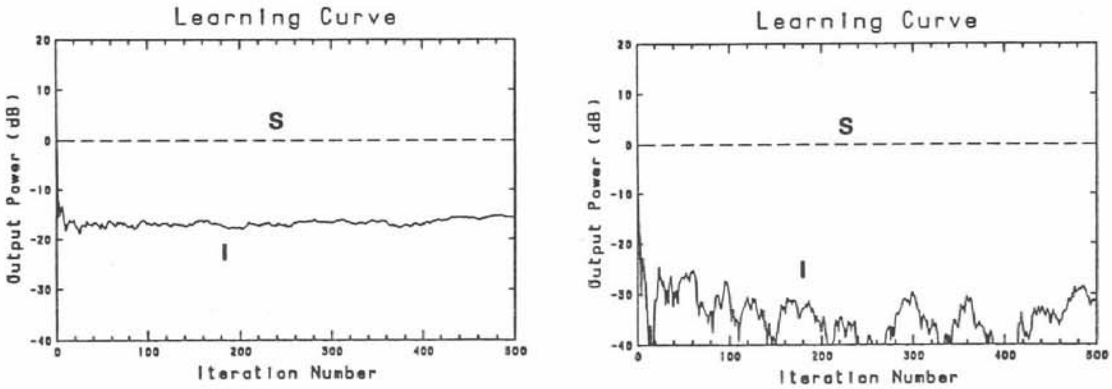
References

[1] K.Takao, T.Fujita, T.Nishi, "An Adaptive Antenna Array Under Directional Constraint," *IEEE Trans. Antennas & Propag.*, vol.AP-24, No.5, pp.662-669, Sept 1976.

[2] J.E.Hudson, "*Adaptive Array Principles,*" Peter Peregrinus, 1981.

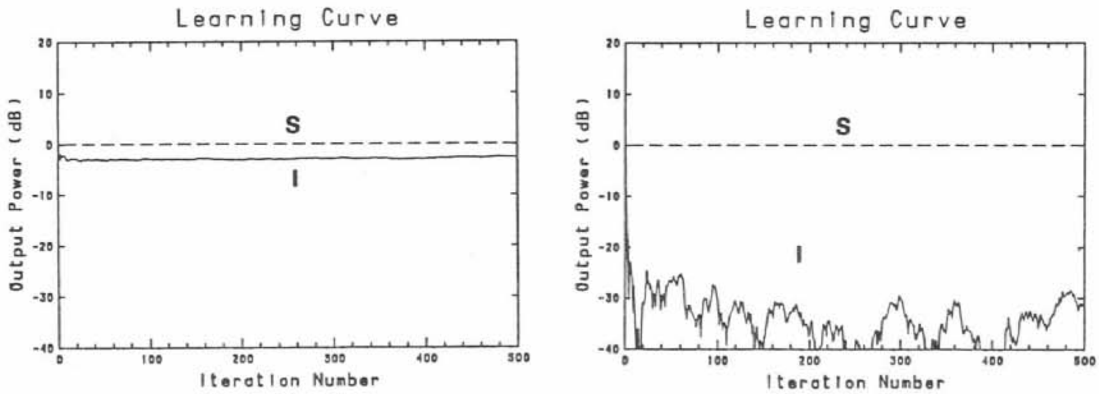
desired signal (S)	$\theta_{\mathrm{S}} = 0°$	$p_{\mathrm{S}} = 1$
interference (I)	$\theta_{\mathrm{I}} = -50°$	$p_{\mathrm{I}} = 100$
thermal noise (N)		$p_{\mathrm{N}} = 0.01$
forgetting factor	$\beta = 0.99$	
weight error (rms)	(a) : 1 (%) (b) : 5 (%)	

Table 1: Input data used in the computation.



(a) (b)

Fig.1 : (error : 1 %)
(a) The learning curves for the SMI.
(b) The learning curves for the recursive algorithm.



(a) (b)

Fig.2 : (error : 5 %)
(a) The learning curves for the SMI.
(b) The learning curves for the recursive algorithm.