Robust Beamforming by an Improved Neural Minor Component Analysis Algorithm

Yabo Yuan, Bin Wu
Beijing Institute of Tracking and Telecommunication Technology
Beijing, China
Email: yaboyuan@gmail.com

***Abstract*—Recently, neural minor component analysis (M-CA) has attracted much attention. It has features of high calculation speed and strong fault tolerant property. In this paper, neural MCA network is applied in adaptive beamforming. In order to get over the disadvantages that MCA based beamformer cannot form null steering beams at the direction of interference and avoid stagnation behavior of algorithm convergence, a novel algorithm combining the least mean square (LMS) algorithm with MCA algorithm is provided, which uses LMS learning rule to continue the training of the neural MCA network. Simulations show that the MCA-LMS based beamformer has strong stability, resembled convergence rates and can obtain a 0dB main-lobe at the direction of primary signal while forming a more than 40dB null steering at the direction of interference.**

***Keywords*-neural network, minor component analysis, least mean square, adaptive beamforming**

I. Introduction

Minor component analysis is a statistical method for extracting eigenvectors associated with the smallest eigenvalue of the covariance matrix of the input data [1]. Recently, there have been much interest in the connection between MCA algorithm and neural network [2] - [3]. Neural network learning rules for neural MCA network update the estimation of the minor directions after each presentation of a data point, which brings advantages for high dimensional data, since the computation of the large covariance matrix can be avoided. For a linear neural unit as depicted in Fig.1, the MCA learning rules allow the vector of synaptic weight to converge towards the eigenvector of the smallest eigenvalue of the input signal's covariance matrix.

Neural MCA network techniques have several applications, such as moving target tracking, adaptive array processing, frequency estimation, signal parameter estimation, *et al.*. Robust beamforming problem is to adapt the spatial filter of a source so that its direction of arrival (DOA) may be available. Fiori applied OJA+MCA neural network to beamforming problem and obtained good performance [4], but its learning algorithm is not globally convergent. FMCA (first minor component analysis) is presented for robust constrained beamforming in [5], where the range of the minor eigenvalue is restricted to ensure convergence. [6] provide a globally convergent learning algorithm for MCA neural network and invites it to beamforming problem. In [7], a novel MCA learning rule is presented for robust constrained beamforming, which is stable independently of the minor eigenvalue. However, the iteration process of MCA algorithms often experience a stagnation, besides, it cannot restrict beamformer response on the direction of interference. In this paper, we get over those shortages by combining the MCA algorithm with LMS algorithm.

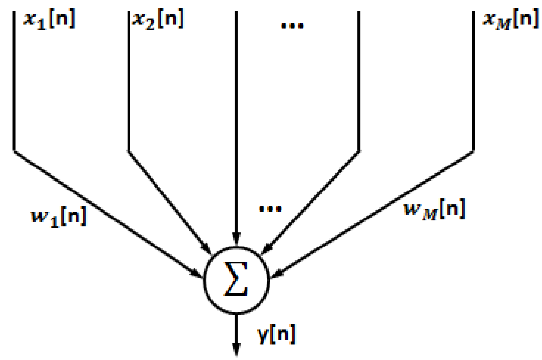


Figure 1. Linear neuron system model.

This paper is organized as follows: first, we present the principle of FMCA algorithm and introduce a penalty term to ensure the stabilization; then we build a MCA based beamformer system model and express its performance metrics; after that, the MCA-LMS algorithm is invited to the beamforming problem which contains the benefits of both sides; finally, computer simulations show that our algorithm significantly outperformed the competitor on beamforming performance and restricted the interference at the same time.

II. First Minor Component Extraction

Consider a simple linear neural unit described by $y(t) = \boldsymbol{w^T}(t)\boldsymbol{x}(t)$, where $\boldsymbol{x} \in \mathbb{R}^M$ is the input vector, $\boldsymbol{w} \in \mathbb{R}^M$ is the weight-vector, and y denotes the neuron's output, as depicted in Fig.1.

By the definition of minor component analysis, we have to find the weight-vector $\boldsymbol{w}$ that minimizes the power $E[y^2]$ of the neuron's output, which may be expressed as:

$$\min_{\boldsymbol{w}\in\mathbb{R}^M} E[y^2], \qquad \boldsymbol{w^T w} = 1 \tag{1}$$

Note that $\boldsymbol{w^T w} = 1$ is the constraint of weight-vector $\boldsymbol{w}$.

978-1-4799-7981-3/14 $31.00 © 2014 IEEE
DOI 10.1109/CSE.2014.56

To this aim the cost function may be considered:

$$C(\boldsymbol{w}) \overset{def}{=} \frac{1}{2}E[y^2] + \frac{\lambda}{2}(\boldsymbol{w^T w} - 1) \tag{2}$$

where λ is the Lagrange multiplier. Its gradient has the expression $\frac{\partial C}{\partial \boldsymbol{w}} = E[y\boldsymbol{x} + \lambda \boldsymbol{w}]$, thus the optimal multiplier may be found by solving:

$$\begin{cases} \boldsymbol{w}^T \dfrac{\partial C}{\partial \boldsymbol{w}} = E[y^2] + \lambda \boldsymbol{w^T w} \\ \quad \boldsymbol{w^T w} = 1 \end{cases} \tag{3}$$

Then we are able to define the steepest descent learning rule:

$$\begin{cases} \dfrac{d\boldsymbol{w}}{dt} = -\dfrac{\partial C}{\partial \boldsymbol{w}} = -E[y\boldsymbol{x} - y^2 \boldsymbol{w}] \\ \quad \boldsymbol{w}(0) = \boldsymbol{w}_0 \end{cases} \tag{4}$$

However, the iteration of (4) results to be unstable [8]. In an effect to overcome this shortage, we include a penalty term $-\overline{\sigma}^2(\boldsymbol{w^T w} - 1)\boldsymbol{w}$ to insure stabilization. where $\overline{\sigma}$ is an arbitrary parameter, thus (4) may be rewritten as:

$$\begin{aligned} \frac{d\boldsymbol{w}}{dt} &= -\frac{\partial C}{\partial \boldsymbol{w}} = -E[y\boldsymbol{x} - y^2\boldsymbol{w}] - \overline{\sigma}(\boldsymbol{w^T w} - 1)\boldsymbol{w} \\ &= -(\boldsymbol{C_x} - \overline{\sigma}\boldsymbol{I})\boldsymbol{w} + \boldsymbol{w}^T(\boldsymbol{C_x} - \overline{\sigma}\boldsymbol{I})\boldsymbol{w}\boldsymbol{w} \\ &= \overline{\boldsymbol{C_x}}\boldsymbol{w} - \boldsymbol{w}^T\overline{\boldsymbol{C_x}}\boldsymbol{w}\boldsymbol{w} \end{aligned} \tag{5}$$

where we let $\boldsymbol{C_x} \overset{def}{=} E(\boldsymbol{x}\boldsymbol{x^T})$, $\overline{\boldsymbol{C_x}} \overset{def}{=} -(\boldsymbol{C_x} - \overline{\sigma}\boldsymbol{I})$. If we set $\overline{\sigma}$ bigger than the largest eigenvalue of $\boldsymbol{C_x}$, $\overline{\boldsymbol{C_x}}$ becomes a non-negative matrix and *Theorem 1* in [5] may be applies to system (5) which ensure the stabilization of the algorithm. According to [5], we rewrite the theorem as:

Theorem Let $\boldsymbol{C_x}$ be the covariance matrix of the random process $\boldsymbol{x}(t)$ with eigenpairs $(\sigma_1, \boldsymbol{q}_1), ..., (\sigma_M, \boldsymbol{q}_M)$. Suppose eigenvalues are distinct and arranged in descending order, eigenvectors are normalized so that $\boldsymbol{q}_k^T\boldsymbol{q}_k = 1$, and $\boldsymbol{w}_0^T\boldsymbol{q}_k \neq 0$. If $\overline{\sigma} > \sigma_1$ then the state vector $\boldsymbol{w}$ of system (5) asymptotically converges towards $+\boldsymbol{q}_M$ or $-\boldsymbol{q}_M$.

III. Application To Robust Beamforming

An on-line beamformer may be realized by a linear neural unit described by a complex weight-vector $\boldsymbol{w} \in \mathbb{C}^M$, where M is the number of sensors. The input-output relationship is $y = \boldsymbol{w}^H\boldsymbol{x}$. By restricting to planar geometry, the array configuration is expressed by a steering vector $\boldsymbol{a}(\theta)$ defined as the vector of phase delays to align the array outputs for a plane wave coming from direction θ. Considering a uniform linear array with M elements spaced by distance d, the steering vector finds to be

$$\boldsymbol{a}(\theta) = [1, e^{-j\frac{2\pi}{\lambda}dsin\theta}, ..., e^{-j\frac{2\pi}{\lambda}(M-1)dsin\theta}]^T \tag{6}$$

where λ is the wavelength. The covariance matrix $\boldsymbol{C_x}$ of the array input signal $\boldsymbol{x}$ is

$$\boldsymbol{C_x} = \sigma_s^2\boldsymbol{a}\boldsymbol{a}^H + \sigma_n^2\boldsymbol{Q} \tag{7}$$

The noise cross-spectral matrix Q is normalized to have its trace equal to the number of sensors M so that σ_s^2/σ_n^2 is the input signal-to-noise spectral ratio averaged across the M sensors. The array beampattern $P_w(\theta)$ represents beamformer response of direction θ owing to beamforming, which can be expressed in vector form as:

$$P_w(\theta) \overset{def}{=} \boldsymbol{w}^H\boldsymbol{a}(\theta) \tag{8}$$

The array gain $G(\theta)$ represents the improvement of signal-to-noise ratio (SNR) along direction θ, that is:

$$G(\theta) \overset{def}{=} \frac{|\boldsymbol{w}^H\boldsymbol{a}(\theta)|^2}{\boldsymbol{w}^H\boldsymbol{N}\boldsymbol{w}} \tag{9}$$

When the noise is spatially white, *i.e.* $\boldsymbol{N} = \boldsymbol{I}_M$, the array gain becomes what is called white noise gain $G_w(\theta) \leqslant M$ [9]. Furthermore, the sensitivity of array gain to signal mismatching by considering the signal $\boldsymbol{x}$ perturbed by small random zero-mean errors, defined as $S(\theta)$, proves to equal $G_w^{-1}(\theta)$ under white noise condition [9]. Prescribing a bound for the sensitivity of the array usually means constraining the white noise gain in the looking direction as $G_w(\theta_s) = \delta^2$, where the value δ^2 must be chosen less or equal to M for the constraint to be consistent, and larger δ provides stronger robustness.

By denoting with θ_s the direction of arrival of the primary source, a constraint writing $\boldsymbol{w}^H\boldsymbol{a}(\theta_s) = 1$ may be set, which ensures no attenuation and zero phase-shift in the expected direction.

A way to train the beamforming neuron is to design a learning rule as a system to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \boldsymbol{w}^H\boldsymbol{C_x}\boldsymbol{w}, \quad \boldsymbol{w^T w} = \delta^{-2}, \boldsymbol{w}^H\boldsymbol{a}(\theta_s) = 1 \tag{10}$$

In [9] an algorithm is proposed to update the weight vector so that the second constraint in (10) is fulfilled while minimizing the spectral power of the neuron's output $C(\boldsymbol{w}) = E[y^2]$. The algorithm can be expressed as

$$\boldsymbol{w}(t+1) = \overline{\boldsymbol{w}} + \overline{\boldsymbol{P}}\boldsymbol{w}(t) - \mu\overline{\boldsymbol{P}}\frac{\partial C}{\partial \boldsymbol{w}(t)} \tag{11}$$

where $\overline{\boldsymbol{P}} \overset{def}{=} \boldsymbol{I_M} - \boldsymbol{a}(\theta_s)(\boldsymbol{a}(\theta_s)^H\boldsymbol{a}(\theta_s))^{-1}\boldsymbol{a}(\theta_s)^H$ is the projection matrix which projects w onto the null space of $\boldsymbol{a}(\theta_s)^H$; $\overline{\boldsymbol{w}} \overset{def}{=} \boldsymbol{a}(\theta_s)(\boldsymbol{a}(\theta_s)^H\boldsymbol{a}(\theta_s))^{-1}$ is the projection matrix which projects $\boldsymbol{w}$ onto the range of $\boldsymbol{a}(\theta_s)$. An extra mechanism is needed for optimization on the first constraint. Here we use instead the FMCA cost function

$$C_{MCA}(\boldsymbol{w}) \overset{def}{=} \frac{1}{2}E[y^2] + \frac{\lambda}{2}(\boldsymbol{w^T w} - \delta^{-2}) \tag{12}$$

According to equation [5], the gradient is:

$$\frac{\partial C_{MCA}(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{C_x}\boldsymbol{w} - \delta^{-2}(\boldsymbol{w}^H\boldsymbol{C_x}\boldsymbol{w})\boldsymbol{w} + \overline{\sigma}(\boldsymbol{w^H w} - \delta^{-2})\boldsymbol{w} \tag{13}$$

where $\overline{\sigma} > \delta^{-2}\overline{\sigma}_1$, $\overline{\sigma}_1$ is the greatest eigenvalue of the covariance matrix. It allows to satisfy both constraints without additional mechanisms.

IV. Combination With LMS Algorithm

The MCA algorithm has the characteristics of rapid convergence, strong fault tolerance etc. However, as the iteration process goes on, the algorithm often show a stagnation phenomenon, and cannot further reduce the value of cost function. Besides, although the MCA algorithm minimizes the output signal power and extracts the minor component of input signal to fulfill the unit boresight response constraint, it cannot distinguish between an expected signal and an interference signal, thus the beamformer may response to the wrong direction. By contrast, LMS algorithm uses the strategy of minimizing mean square error between output signal and the expected signal (input signal on the direction of arrival), which guarantee the right direction for beamforming. The cost function may be expressed as:

$$C_{LMS}(\boldsymbol{w}) = E[(d - \boldsymbol{w}^H \boldsymbol{x})^2] \tag{14}$$

where d stands for the expected signal. LMS based beamformer can force the main beam to direct at the direction of desired user and null steering beams at the direction of interference in order to suppressing or canceling the interfering signals. But the LMS algorithm is greatly influenced by initial value and the convergence is quite slow, besides it requires high SNR [10]. In this section, we invite a MCA-LMS algorithm which combines the two algorithms together. First we use MCA learning rule to train the weight vector of the beamformer to direct the main beams to incoming signal directions, then we invite LMS strategy to suppress beams on the interference directions so that the interference can be restricted and the main beam may be quickly directed to the desired direction.

Algorithm 1 illustrates the details of the proposed scheduling algorithm.

Algorithm 1 MCA-LMS ALGORITHM

1: $\boldsymbol{w}_1(0) = \boldsymbol{w}_0, \boldsymbol{w}_0 \neq 0, k = 0;$
2: **if** $k < n_1$ **then**
3: $\quad \boldsymbol{w}(k+1) = \overline{\boldsymbol{w}} + \overline{\boldsymbol{P}}\boldsymbol{w}(k) - \eta \overline{\boldsymbol{P}} \frac{\partial C_{MCA}(\boldsymbol{w}(k))}{\partial \boldsymbol{w}(k)};$
4: $\quad k = k+1;$
5: **end if**
6: **if** $k < n$ **then**
7: $\quad \boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \mu \boldsymbol{P} \frac{\partial C_{LMS}(\boldsymbol{w}(k))}{\partial \boldsymbol{w}(k)};$
8: $\quad k = k+1;$
9: **end if**

V. Simulation Results

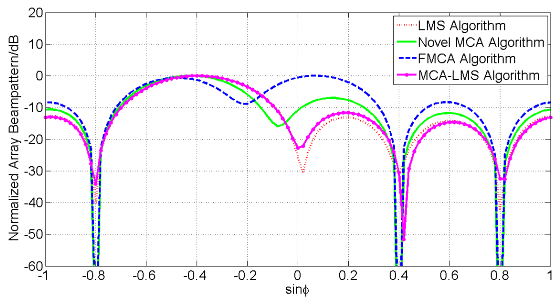
Table I
SIMULATION PARAMETERS

M, Array Number	5
n, Number of snapshots	100
θ_s, Direction of arrival	$\arcsin(-0.4)$
θ_s, Direction of interference	0
TNR, Target-to-noise ratio	10/-10/-40dB
INR, Interference-to-noise ratio	10dB
δ,	1.5
μ,	0.001
$\overline{\sigma}$,	$1.5trace(\boldsymbol{C_x})$
η, Step size	0.0005
n_1,	10
n,	100
m, Number of simulations for statistical accuracy	1000

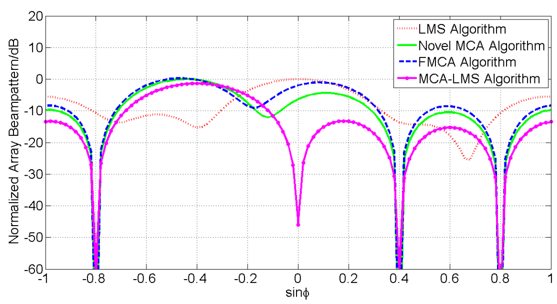
We perform Monte Carlo simulations on the MCA-LMS algorithm, and compared it with LMS, FMCA and a Novel MCA algorithm mentioned in [7], which includes a penalty term on the self-stabilizing MCA learning rule. The performance of the beamformer in a weak source signal and strong white noise situation is illustrated in the simulations. The uniform linear array consists of five elements spaced half a wavelength apart. Simulation parameters are presented in *Table 1*.

Fig.2 shows the normalized array beampatterns under different TNR after learning. We can see that LMS based beamformer can only direct to the expected direction when TNR is high (Fig. 2(a)). When TNR is low (Fig. 2(b),(c)), it can barely realize beamforming on the target direction after 100 times of iteration. Actually, experimentation proofs that after 10000 iteration, the result is still poor. This is because LMS algorithm is greatly influenced by initial value and the convergence rate is limited by the step size. The array beampattern of FMCA and Novel MCA algorithm peak at the target direction as well as the interference direction especially when the target signal are comparable to the interference signal (Fig. 2(a)), because they cannot distinguish between the two arriving signals, thus they may enhance the interference. The MCA-LMS based beamformer can obtain a 0dB main-lobe at the DOA of the primary signal and significant attenuation in the other zones of the space, especially at the interference direction where a null steering (about 20dB when TNR=10dB and 46dB when TNR=-10/-40dB) is achieved. It has resembled strong signal tracking ability.

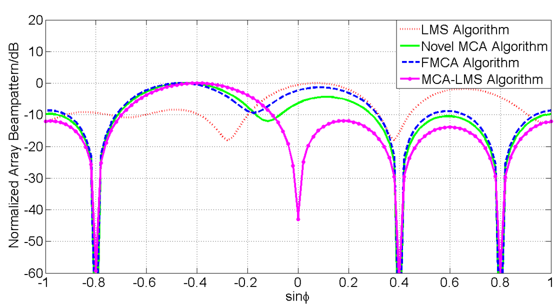
Fig.3 present the array gain as a function of the number of iterations. It is shown after 100 times of iteration, the MCA-LMS algorithm achieves the largest array gain (about 7dB), which means that it provides the biggest improvement of SNR along direction θ_s. Array gain of comparing algorithms are less than 6dB, with the LMS algorithm about -4dB because of the less of iteration. Fig.4 illustrates the trend of cost functions. It can be seen that the MCA algorithm



(a) TNR = 10dB.



(b) TNR = -10dB.



(c) TNR = -40dB.

Figure 2. Normalized array beampattern after 100 times of iteration.

stagnates at the second iteration, while the stagnation is broken and the value of cost function continues to decrease to zero with the help of LMS iterations in the MCA-LMS algorithm.

VI. Conclusion

Neural minor component analysis has the features of high calculation speed and strong fault tolerant property. We applied neural MCA network to adaptive beamforming problem and introduced a penalty term to ensure the stabilization. In order to get over the disadvantages that MCA based beamformer cannot form null steering beams at the direction of interference and to avoid stagnation behavior of algorithm convergence, we combined the LMS algorithm with MCA algorithm, and used LMS learning rule to continue the training of the neural MCA network. We performed Monte Carlo simulations on the MCA-LMS algorithm, and compared it with LMS, FMCA and a Novel MCA algorithm. Computer simulations showed that the MCA-LMS based beamformer has strong stability, resembled convergence rates and can obtain a 0dB main-lobe at the direction of primary signal while forming a more than 40dB null steering at the direction of interference.

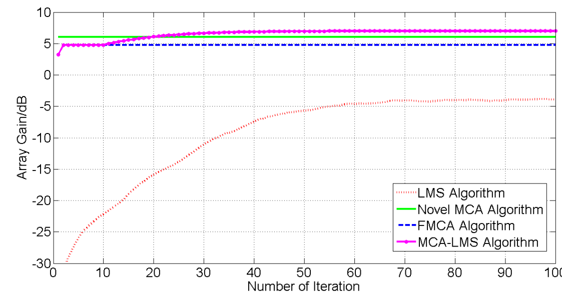


Figure 3. Array gain versus iteration.

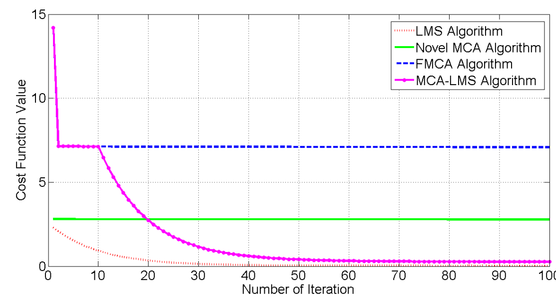


Figure 4. Cost function value versus iteration.

References

[1] E. Oja, “Principal components, minor components, and linear neural networks,” *Neural Networks*, vol. 5, no. 6, pp. 927–935, 1992.

[2] G. Cirrincione, M. Cirrincione, J. Hérault, and S. Van Huffel, “The mca exin neuron for the minor component analysis,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 1, pp. 160–187, 2002.

[3] R. Möller, “A self-stabilizing learning rule for minor component analysis,” *International journal of neural systems*, vol. 14, no. 01, pp. 1–8, 2004.

[4] S. Fiori and F. Piazza, “Neural mca for robust beamforming,” in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 3. IEEE, 2000, pp. 614–617.

[5] S. Fiori, “Neural minor component analysis approach to robust constrained beamforming,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 4, pp. 205–218, 2003.

[6] M. Ye, "Robust beamforming by a globally convergent mca neural network," in *Advances in Neural Networks–ISNN 2005*. Springer, 2005, pp. 1034–1041.

[7] D. Tian, J. Wang, Y. Xue, and G. Xue, "A neural minor component analysis algorithm for robust beamforming," in *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, vol. 2. IEEE, 2005, pp. 1182–1185.

[8] E. Oja and L. Wang, "Robust fitting by nonlinear neural units," *Neural networks*, vol. 9, no. 3, pp. 435–444, 1996.

[9] H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 10, pp. 1365–1376, 1987.

[10] A. Hakam, R. Shubair, S. Jimaa, and E. Salahat, "Robust interference suppression using a new lms-based adaptive beamforming algorithm," in *Mediterranean Electrotechnical Conference (MELECON), 2014 17th IEEE*. IEEE, 2014, pp. 45–48.