Minor Component Analysis with Implementation to Blind 2-channel Equalization

Jie Luo, Xieting Ling
Fudan University, Shang Hai, 200433, P.R.China
email: xtling@fudan.ihep.ac.cn

Abstract

A null space method is proposed for blind identification of non-minimum phase linear time-variant channels using minor component analysis. The algorithm is proved to be globally convergent in discrete domain. The fast convergence speed makes this algorithm very useful for tracing a time-variant channel in mobile communication. Some simulations are given to show its tracking ability, and a very efficient method for equalizing the blind channel is also proposed at the end of this paper.

1. Introduction

In order to overcome the non-minimum phase problem, conventional approaches for blind-channel identification are naturally addressed using high order cumulant (HOC) of the signal. Such HOC based methods, as expected, suffer from computational intensity, unreliability of high order statistics, and slow convergence rate. Recently, Dong et al[1] proposed a new method using two receivers and a neural network with orthogonal learning rule for blind channel identification. This algorithm is proved to be globally convergent in the contiuous domian. However, due to the difference between the continuous domain and the discrete domain, computer simulations show that divergence occurs when signal amplitude is relatively large or the learning step is not very small. Meanwhile, the algorithm needs a precise estimation of the channel order to design the neural network. This problem is vital but no discussion was given to it in [1].

In this paper, a new null space learning rule based on the same network of [1] is proposed. The algorithm is proved to be much faster than which of [1] and is globally convergent in the discrete domain. The problem of channel order determination has also been solved in this paper. Simulation results are given to show its tracking ability and a very efficient method to equalize the blind channel is presented at the end of this paper.

2. Network structure

Suppose the linear communication channels can be described as FIR filters, the structure of the network proposed in [1] is shown in Figure 1.

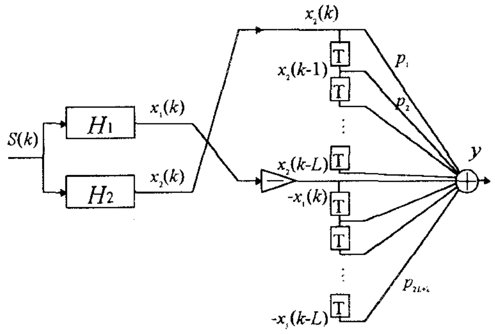

Figure 1. Structure of the network proposed in [1]

Here $S(k)$ represents the common input signal sequence, $H_j(j=1,2)$ describes the jth communication channel, and $x_j(k)\,(j=1,2)$ is the output sequence of the jth channel received by the jth sensor. $p_1 \; \dots \; p_{2L+2}$ represent the weights of the neural network whose output is y. Then we can describe the channels' transfer function in the domain of Z as

$$H_j(z)=\sum_{n=0}^{L} h_j(n)z^{-n} \qquad (j=1,\,2) \tag{1}$$

where L is the higher order of the two channels. Here we just suppose we have known the order exactly. We define two arrays

$$\vec{X}_k=\left[x_2(k),\,\dots x_2(k-L),\,-x_1(k),\,\dots -x_1(k-L)\right]^T$$
$$\vec{H}=\left[h_1(0),\,\dots h_1(L),\,h_2(0),\,\dots h_2(L)\right]^T$$

in space R^{2L+2}, and suppose the input signal $S(k)$ to be random, we can reach the following theorem

Theorem 1: The correlation matrix of $\vec{X}_k$ will have a distinctive zero eigenvalue whose corresponding eigenvector will just be the normalized vector of $\vec{H}$, if the following conditions are satisfied

1. $H_1(z)$ and $H_2(z)$ have no common zeros.

2. At least the higher order of the channels is L.

Proof: First, in the aspect of existence, We can describe the relationship between $x_i(k)$ and $S(k)$ as

$$\begin{cases} x_1(k) = \sum_{j=0}^{L} s(k-j)h_1(j) \\ x_2(k) = \sum_{j=0}^{L} s(k-j)h_2(j) \end{cases} \tag{2}$$

If the correlation matrix of $\vec{X}_k$ is denoted as R,

$$R\vec{H} = E\left[\vec{X}_k \vec{X}_k^T\right]\vec{H} = E\left[\vec{X}_k \vec{X}_k^T \vec{H}\right] \equiv \vec{0} \tag{3}$$

can be obtained, that means the normalized vector of $\vec{H}$ is an eigenvector of R and the corresponding eigenvalue of it is 0.

Second, in the aspect of uniqueness. Considering the following equation

$$\vec{X}_k = \begin{bmatrix} h_2(0) & h_2(1) & \cdots & 0 & \cdots & 0 \\ 0 & h_2(0) & \cdots & h_2(L) & \cdots & 0 \\ 0 & \cdots & h_2(0) & \cdots & \cdots & h_2(L) \\ -h_1(0) & -h_1(1) & \cdots & 0 & \cdots & 0 \\ 0 & -h_1(0) & \cdots & -h_1(L) & \cdots & 0 \\ 0 & \cdots & -h_1(0) & \cdots & \cdots & -h_1(L) \end{bmatrix} \begin{bmatrix} S(k) \\ S(k-1) \\ \vdots \\ \vdots \\ \vdots \\ S(k-2L) \end{bmatrix} \tag{4}$$

it has been proved that when condition 1,2 is satisfied, the transfer matrix will have full column rank[2]. Since $S(k)$ is random, the space spanned by $\vec{X}_k$ will have a dimension of $2L+1$, indicating that the zero eigenvalue is distinctive. Proof completed.

Considering R is quasipositive definite, we can easily find out that the zero eigenvalue is the minimum eigenvalue of matrix R and the normalized vector of $\vec{H}$ is just the minor component of R.

3. Learning algorithm and convergence discussion in discrete domain

As has been mentioned by Oja in [3], the convergence proof in the continuous domain will sometimes not be true in the discrete domain. Similar to the appendix of [3], in this section, we will propose our learning algorithm in the discrete domain and then give the proof of its global convergence.

Suppose $0 < b \le 1$, we define $\xi = \dfrac{|\vec{P}(t)|^2 - 1}{|\vec{P}(t)|^2 + 1}$, $q = |\vec{X}(t)|^2 - \dfrac{y^2}{|\vec{P}(t)|^2}$. Since $\xi < 1$ and $q \ge 0$, $\dfrac{(1-\xi)}{q} > 0$ can always be satisfied. Then our learning rule can be written as

$$\vec{P}(t+1) = \vec{P}(t) + b\left(\frac{1-\xi}{q}\right)\left(-y\vec{X}(t) + \frac{y^2}{|\vec{P}(t)|^2}\vec{P}(t)\right) - \xi\vec{P}(t) \tag{5}$$

It can be easily proved out that all the normalized eigenvector of R are the equilibrium points of (5), we will prove that only the normalized minor component of R is stable.

First, in the aspect of direction, we denote the angle between $\vec{P}(t)$ and $\vec{X}(t)$ in R^{2L+2} as $\theta(t)$, and the angle between $\vec{P}(t+1)$ and $\vec{X}(t)$ as $\theta(t+1)$, define

$$\rho(b) = \frac{\cos^2\theta(t+1)}{\cos^2\theta(t)} = \frac{|\vec{P}^T(t+1)\vec{X}(t)|^2 \big/ |\vec{P}(t+1)|^2}{|\vec{P}^T(t)\vec{X}(t)|^2 \big/ |\vec{P}(t)|^2} \tag{6}$$

and considering $y^2 = |\vec{P}(t)|^2|\vec{X}(t)|^2\cos^2\theta(t)$, we can get from (5)

$$\rho(b) = \frac{|\vec{P}(t)|^2(1-\xi)^2(1-b)^2}{(1-\xi)^2\left[|\vec{P}(t)|^2 + b^2y^2/q\right]} \le \frac{|\vec{P}(t)|^2(1-b)^2}{|\vec{P}(t)|^2} = (1-b)^2 < 1 \tag{7}$$

that is to say $\theta(t) \to \frac{\pi}{2}$ can be ensured for every learning step. Especially when $b=1$, $\rho(b)=0$ can be got for every input vector in just one step, indicating that the network will finally converge to the minor component in the aspect of direction.

In the aspect of amplitude, we have

$$\begin{aligned} |\vec{P}(t+1)|^2 &= \frac{4|\vec{P}(t)|^2}{\left(|\vec{P}(t)|^2+1\right)^2}\left(1 + b^2ctg^2\theta(t)\right) \\ &\le \frac{4|\vec{P}(t)|^2}{\left(2|\vec{P}(t)|\right)^2}\left(1 + b^2ctg^2\theta(t)\right) = 1 + b^2ctg^2\theta(t) \end{aligned} \tag{8}$$

This gives the upper bound of amplitude. Then let us consider the lower bound. If $|\vec{P}(t)|^2 < 1$, from (8), we have

$$|\vec{P}(t+1)|^2 > |\vec{P}(t)|^2\left(1 + b^2ctg^2\theta(t)\right) \tag{9}$$

indicating the amplitude has a tendency of increasing. Meanwhile, when $1 + b^2ctg^2\theta(t) \ge |\vec{P}(t)|^2 \ge 1$, due to (8),

$$|\vec{P}(t+1)|^2 \ge \left(1 + b^2ctg^2\theta(t)\right) \big/ |\vec{P}(t)|^2 \ge 1 \tag{10}$$

can be obtained. In fact, this insure us that the lower bound of the amplitude will finally be 1. Since $\theta(t) \to \frac{\pi}{2}$, $\left(1 + b^2ctg^2\theta(t)\right) \to 1$ surely holds, accordingly affirming the convergent point of the amplitude to be 1. Proof completed.

4. Channel order determination

Though we have prove the global convergence, there is still one problem left unsolved. Accurately estimating the channel order is vital to our method but in (4) we just supposed it to be L. Some HOC based methods for order determination have already been proposed, but computer simulations show they often fail to insure a precise order estimation. In this section, we will present an approach not appealing in appearance but very efficient to determine the higher order of the channels.

Suppose we have two neural networks, each is the same as the one shown in figure 1. The two networks are both designed according to an arbitrary order L, and their inputs are the same ($\vec{X}_k(t)$). The only difference between them is the initial condition. On one hand, if the supposed order L is higher than the actual one, from (4) we can see the minor component of R will no longer be distinctive. Since the two neural networks are different in initial condition, their learning result will also be different. So the correlation value of the two weight vectors will have a large probability to be small, at least not too high to be above 0.95 for example. On the other hand, if the supposed order L is lower, $\vec{X}_k$ will then span the entire space of dimension $2L+2$. Then the minor eigenvalue of R will no longer be zero, but most probably will remain distinctive. As we have discussed before, the two networks will still converge to the distinctive minor component. Their crosscorrelation of weight vectors will keep a very high value close to 1. From this phenomenon, whether the arbitrary order is higher or not can be easily judged. That will be enough for us to find the proper order.

We design the two neural networks in an arbitrary high order first, then let them have a downward search to find the proper channel order. Meanwhile, every time when the current order L of the two NN is not higher than that of the real channel, we increase the design order to $L+1$. Thus, if the higher order of the physical channels is L_0, the design order of the two NN will finally oscillate between L_0 and L_0+1. Of course, the proper order can be picked out easily. Because of the avoiding of high order statistics caculation and the high convergence speed of the algorithm, our order determination method is very accurate and efficient to be put into practice. Furthermore, it can even trace the channel order varying in a relatively low speed.

5. Final structure and channel equalization

With the discussion in section 3 and 4, we can present our final structure of blind channel identification in figure 2. Here without losing the high convergence speed, we add a moving average filter behind each neural network to reduce the effect of noise on the output. In figure 2, NN1 is used to identify the channel coefficients while NN2 and NN3 are used to estimate the channel order.

Next, we will propose our decision-feedback approach to equalize the identified channel. Suppose we have

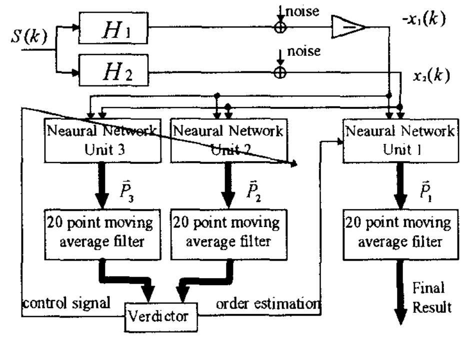

Figure 2. Structure of the network in practice ($\vec{P}_1$, $\vec{P}_2$, $\vec{P}_3$ respectively represent the weight vector of the neural networks)

known the signal sequence sent before to be $\vec{S}_{k-L}$, $\vec{S}_{k-L} = [s(k-1) \;\ldots\; s(k-L)]^t$, and we have got the channel coefficient to be $\vec{H}_1$, $\vec{H}_1 = [h_1(0) \;\ldots\; h_1(L)]^t$. What we do not know is just the current one step of the signal, which we suppose to be $s_g(k)$, then $\vec{H}_1{}^t \begin{bmatrix} s_g(k) \\ \vec{S}_{k-L} \end{bmatrix} = x_1(k)$ must holds true. That means

$$s_g(k) = \frac{1}{h_1(0)}\left[x_1(k) - \vec{H}_1{}^t \begin{bmatrix} 0 \\ \vec{S}_{k-L} \end{bmatrix}\right] \tag{11}$$

from this equation, the right signal can be easily found.

In practice, however, the channels and signals are all blind to us, it is improbably to get $\vec{S}_{k-L}$ correctly. Fortunately, computer simulations show that we can initiate the first vector $\vec{S}_0$ with a random vector, and with the identified channel coefficients $\vec{H}_1$, the equalization network will find the right answer in just several steps. The equalization structure is shown in figure 3.

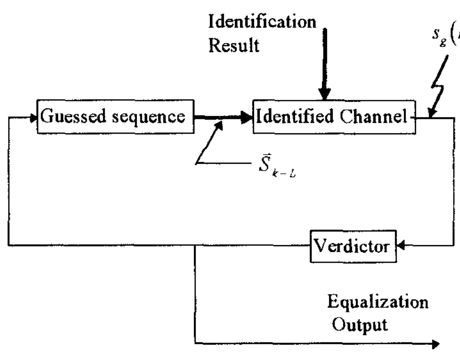

Figure 3. Equalization structure

6. Simulation results

In this section, two simulations are given to show the tracking ability and equalizing ability of our network. In each simulation, we initiate the weights of the neural networks with a normalized random vector, and let $b = 1$ to set the convergence rate at its fastest point. For convenience, we omit the order determination course in the following simulations.

Simulation 1: Here the inputs $\bar{X}(t)$ of the neural networks are outputs of two channels with PAM scheme driven by a randomly distributed 4-ary transmitting symbols $s(k)$. The tracking ability of the network is simulated by setting the symbol rate at 300kHz with channel Doppler spread 100Hz, corresponding to a speed of a mobile receiver up to about 70mph. For PAM signals, the time-varying channel coefficients are simulated by low frequency Guassian noise with the bandwidth equal to the Doppler spread. The trajectories of the first three coefficients of the first channel are given by the dotted lines in figure 4, while their estimations are given by the solid lines.

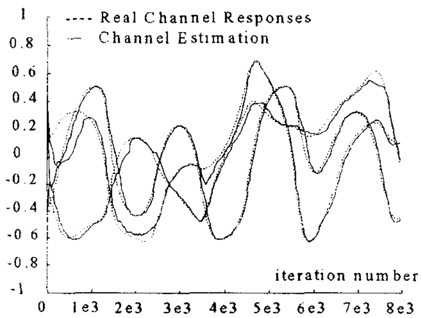

Figure 4. The trajectories of channel

Simulation 2: The order of this simulation is to show the equalizing ability of the network. This time the transmitting symbols $s(k)$ is randomly distributed 2-ary sequence. The actual channel impulse responses are set as $\bar{H}_1 = [1.0 \quad 1.5 \quad 0.7 \quad -0.4]^T$, $\bar{H}_2 = [1.0 \quad 0.9 \quad 0.6 \quad 0.2]^T$. We reduce SNR of the receivers to 14db, figure 5 show the eye graph of the verdictor input described in equation (5). Although a precise channel identification can no longer be obtained with low SNR, we can still get a good equalization result which indicate the high anti-noise capacity of the equalizer.

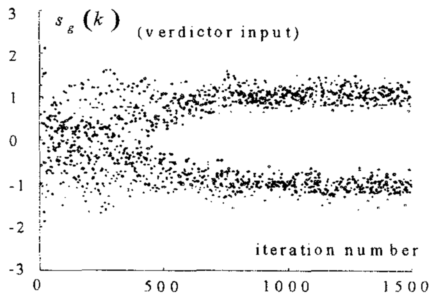

Figure 5. Eye graph of verdictor input

7. Conclusions

A new method for blind 2-channel equalization using minor component analysis is proposed. Proof and computer simulations show its high convergence speed, low computational complicity and high anti-noise capacity. It is very efficient in equalizing time-variant MA channels in mobile communication.

8. References

[1] Guojie Dong, Ruey-wen Liu, "An Orthogonal Learning Rule for Null-Space Tracking with Implementation to Blind 2-Channel Identification", *Technical Report, University of Notre Dame*, December 1995.

[2] E. Moulines, P. Duhamel, J-F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters", *IEEE Trans. Signal processing*, Vol. 43, 1995, pp: 516-525.

[3] Lei Xu, Krkki Oja, and Ching Y. Suen. "Modified Hebbian Learning for Curve and Surface Fitting," *Neural Networks*, Vol 5, 1992, pp: 441-457.