COMPLEX DOMAIN FLEXIBLE NON-LINEAR FUNCTION FOR BLIND SIGNAL SEPARATION

Simone Barbabella, Francesco Piazza[1] *and Aurelio Uncini*[2]

[1] Electronics and Automatics Dept. - University of Ancona
Via Brecce Bianche, 60131 Ancona-Italy
Email: upf@ieee.org - URL: http://nnsp.eealab.unian.it/

[2] INFOCOM Dept. - University of Rome “La Sapienza”
Via Eudossiana, 18, 00183 Roma-Italy
Email: aurel@ieee.org - URL: http://infocom.uniroma1.it/aurel

ABSTRACT

In this paper a new adaptive non-linear function for blind complex domain signal processing is presented. It is based on a couple of spline functions, one for the real and one for the imaginary part of the input, whose control points are adaptively changed using gradient-based techniques. B-splines are used, because they allow to impose only simple constraints on the control parameters in order to ensure a monotonously increasing characteristic.
This new adaptive function is then applied to the outputs of a one-layer neural network in order to separate complex signals from mixtures by maximizing the entropy of the function outputs. We derive a simple form of the adaptation algorithm and present some experimental results that demonstrate the effectiveness of the proposed method.

1. INTRODUCTION

In the last years the blind source separation problem has raised a great interest in the signal processing community. In particular, information maximization techniques, implemented in neural like architecture, have been particularly studied.

In one of the first papers, Bell and Sejnowski [1], proposed the use of a one-layer neural network in order to separate linear mixtures of signals. This architecture is composed by an invertible linear transformation followed by bounded, monotonously increasing, nonlinear functions applied to all outputs separately. The adaptation (or learning in neural network contest) is carried-out by maximizing the output entropy. In this case, if the pdf’s of the sources are known, the fixed non-linearity’s should be taken equal to the cumulative density functions of the sources.

Recently, the problem of source separation has been extended also in the complex case (see for example [2,3]), i.e. in the case of mixture of complex-valued signals, which can have several practical applications.

In both the real and complex cases, however, the cumulative density functions of the sources are usually unknown. Although simulations often exhibit good results also for non-linearity’s that don’t exactly match the signals, in the general case a better estimation of the correct non-linear functions can be important. Therefore several approaches have been proposed in the real case to obtain adaptive non-linearity’s (see for example [4,5]), which allow to optimize the shape of the functions with respect to the input signals.

Among the different approaches, the case of spline-based non-linear functions [5] seems to be particularly appealing. It is based on the idea of deriving adjustable non-linear functions by using a spline approximation whose control points are adaptively changed. This idea was firstly proposed in the supervised context for multilayered networks [6] and then proposed in an information maximization scheme [5].

Recently, a complex-valued adaptive spline neural network has been presented [7]. This architecture is shown to be well suitable for supervised signal processing applications, being based on an efficient Catmul-Rom spline activation function.

Based on the result in [5] and [7], in this paper a new adaptive non-linear function for blind complex domain signal processing is presented. It is based on a couple of spline functions, one for the real and one for the imaginary part of the input, whose control points are adaptively changed using gradient-based techniques. It uses B-spline functions that allow to impose only simple constraints on the control parameters in order to ensure the needed monotonously increasing characteristic. In particular, the problem of adaptively maximizing the entropy of the outputs of a one-layer neural network is considered, in order to separating complex signals from instantaneous mixtures of them.

In Section 2 the new function is presented, while in Section 3 the blind separation application is described. Some experimental results are reported in Section 4.

2. THE COMPLEX DOMAIN B-SPLINE NON-LINEAR FUNCTION

For a better understanding, let’s briefly introduce the real-valued case. The real spline activation functions (see [6]) are smooth parametric curves, divided in multiple tracts (spans) each controlled by four control points. Let *f*(*x*) be the non-linear function to reproduce, then the spline activation function can be expressed as:

$$y = f(x) = \sum_{i=1}^{N-3} f_i(u, \mathbf{Q}_i) \qquad (1)$$

i.e. as a composition of (*N*-2) spans $f_i(u)$ *i*=1,…,*N*-3 (where N is the total number of the control points $\mathbf{Q}_i$ *i*=1,…,*N*) each depending from a local variable $u \in [0,1)$ and controlled by four $\mathbf{Q}_i = \{Q_i, Q_{i+1}, Q_{i+2}, Q_{i+3}\}$ control points (see Fig. 1). The two parameters *i*, *u* can be derived by an internal variable *z*

0-7803-6685-9/01/$10.00©2001 IEEE

$$z = \frac{x}{\Delta x} + \frac{N-1}{2} \qquad (2)$$

$$z = \begin{cases} 1 & \text{if } z < 1 \\ z & \text{if } 1 \le z \le N-3 \\ N-3 & \text{if } z > N-3 \end{cases} \qquad (3)$$

where Δx is the fixed distance between two adjacent control points; the constraints imposed by equation (3) are necessary to keep the input within the active region that encloses the control points. Separating z into the integer and fractional parts using the floor operator $\lfloor . \rfloor$, finally we get

$$i = \lfloor z \rfloor ; \qquad u = z - i \qquad (4)$$

In matrix form the output for the k-th spline span, can be expressed as

$$y_i = f_i\left(u, \mathbf{Q}_{i_u}\right) = \mathbf{T}_u \cdot \mathbf{M} \cdot \mathbf{Q}_{i_u} \qquad (5)$$

where:

$$\mathbf{T}_u = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \qquad (6)$$

$$\mathbf{M} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \qquad (7)$$

with $0 \le u < 1$ and $\mathbf{M}$ is the coefficient matrix of the B-spline version of [6], as in [5].

In the complex domain, we use two distinct real-valued spline functions [7], one for the real part and one for the imaginary part. Therefore equation (5) can be redefined as:

$$\begin{aligned} y_i &= h_{i,R}\left(u_{iR}, \mathbf{Q}_{i_{uiR}}\right) + j \cdot h_{i,I}\left(u_{iI}, \mathbf{Q}_{i_{uiI}}\right) = \\ &= \mathbf{T}_{uR} \cdot \mathbf{M} \cdot \mathbf{Q}_{i_{uiR}} + j \cdot \mathbf{T}_{uI} \cdot \mathbf{M} \cdot \mathbf{Q}_{i_{uiI}} \end{aligned} \qquad (8)$$

where y_i is the complex output of the function and j denotes the imaginary unit. The corresponding structure is shown in Fig. 2. In order to ensure the monotonously increasing characteristic, we must impose the constraint: $Q_1 < Q_2 < \ldots < Q_N$, for both the real and imaginary parts.

3. BLIND SEPARATION OF COMPLEX SIGNALS

To separate complex input signals from instantaneous mixtures, here we follow an information theoretic approach. The entropy of the outputs $y_i(t)$ is maximized with respect to both the parameters of the un-mixing complex matrix $\mathbf{W}$ and the control points of the B-spline complex activation functions. The cost function to be maximized is hence defined as:

$$\begin{aligned} F(\mathbf{W}, \mathbf{Q}_R, \mathbf{Q}_I) &\triangleq H(\mathbf{y}) = E\left\{\ln\left[p_y(\mathbf{y})\right]\right\} = \\ &= E\left\{\ln\left|\mathbf{J}\right|\right\} + H(\mathbf{x}) \end{aligned} . \qquad (9)$$

where the arrays $\mathbf{Q}_R$ and $\mathbf{Q}_I$ indicate respectively the real and imaginary parts of all the B-spline control points, p_y $(\mathbf{y})$ is the complex domain pdf of the outputs $\mathbf{y}$ and $\mathbf{J}$ represents the Jacobian. The joint entropy $H(\mathbf{y})$ is defined as:

$$H(\mathbf{y}) = -\int p_y(\mathbf{y}_R, \mathbf{y}_I) \ln\left(p_y(\mathbf{y}_R, \mathbf{y}_I)\right) d\mathbf{y}_R d\mathbf{y}_I \ . \qquad (10)$$

In the complex domain the adaptation rule for the weights $\mathbf{W}$ can be expressed as:

$$\Delta \mathbf{W}_k = m\left\{\left[\mathbf{W}_{k-1}^H\right]^{-1} + \mathbf{h}(\mathbf{u})\mathbf{x}^H\right\}; \qquad \text{Bell } et\ al. \text{ [1],} \qquad (11)$$

$$\Delta \mathbf{W}_k = m\left\{\mathbf{I} + \mathbf{h}(\mathbf{u})\mathbf{u}^H\right\}\mathbf{W}_{k-1}^H; \qquad \text{Amari [8];} \qquad (12)$$

where the term $\mathbf{h}(\mathbf{u})$, is defined in [8] and implicitly assumed in [1]. Considering the activation function defined in eq. (8), the j-th element of $\mathbf{h}(\mathbf{u})$, assumes the form:

$$\begin{aligned} h_j(u_j) &= \frac{1}{y'_{jR}(u_{jR})}\frac{\partial y'_{jR}(u_{jR})}{\partial u_{jR}} + j\frac{1}{y'_{jI}(u_{jI})}\frac{\partial y'_{jI}(u_{jI})}{\partial u_{jI}} = \\ &= \frac{1}{\Delta u_{jR}}\frac{\ddot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}}{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}} + j\frac{1}{\Delta u_{jI}}\frac{\ddot{\mathbf{T}}_{u_{jI}}\mathbf{M}\mathbf{Q}_{i_{u_{jI}}}}{\dot{\mathbf{T}}_{uI}\mathbf{M}\mathbf{Q}_{i_{u_{jI}}}} \end{aligned} ; \qquad (13)$$

where for the sake of simplicity we assume Δu_{jR} equal to Δu_{jI} (as in practical cases).

In order to adapt the control points of the activation functions, first it is possible to show that the maximization of the functional (9) is equivalent to the maximization of the following quantity:

$$\sum_{j=1}^{N}\ln\left(y'_{jR}\right) + \sum_{j=1}^{N}\ln\left(y'_{jI}\right) + \ln\left(\det\tilde{\mathbf{W}}\right) \qquad (14)$$

where $\tilde{\mathbf{W}}$ is a suitable matrix depending of the un-mixing matrix $\mathbf{W}$ which does no depend on the control points. Then, starting from this, it is possible to demonstrate that the adaptation rule for the control points of the real part function can be expressed as:

$$\begin{aligned} \Delta Q_{i_{u_{jR}}+m} &\propto \frac{\partial\left[\sum_{k=1}^{N}\ln\left(y'_{kR}\right) + \sum_{k=1}^{N}\ln\left(y'_{kI}\right)\right]}{\partial Q_{i_{u_{jR}}+m}} = \frac{\partial \ln\left(y'_{jR}\right)}{\partial Q_{i_{u_{jR}}+m}} = \\ &= \frac{1}{y'_{jR}}\frac{\partial y'_{jR}}{\partial Q_{i_{u_{jR}}+m}} = \frac{\Delta u_{jR}}{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}}\frac{\partial\left(\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}\right)}{\Delta u_{jR}\partial Q_{i_{u_{jR}}+m}} = \frac{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}_m}{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}} \end{aligned} ;$$

A similar expression can be derived for the control points of the imaginary part function, so we can write the final adaptation rule as:

$$\begin{aligned} \Delta Q_{i_{u_{jR}}+m} &= h\frac{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}_m}{\dot{\mathbf{T}}_{u_{jR}}\mathbf{M}\mathbf{Q}_{i_{u_{jR}}}} \\ \Delta Q_{i_{u_{jI}}+m} &= h\frac{\dot{\mathbf{T}}_{u_{jI}}\mathbf{M}_m}{\dot{\mathbf{T}}_{u_{jI}}\mathbf{M}\mathbf{Q}_{i_{u_{jI}}}} \end{aligned} . \qquad (15)$$

where the term h represents the constant adaptation rate for the B-splines control points. For the sake of brevity, the two derivations are not reported here.

3. EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of the proposed method, several different experiments have been carried out. As a numerical example of complex signal separation, an input $\mathbf{x}\in C^4$ formed by a linear mixture of four independent signals arranged in a vector $\mathbf{s}\in C^4$, is choosen. Signal s_1 is 4QAM, s_2 is 16QAM, signal s_3 is a uniform random noise while s_4 is PSK. Each sample is, also, corrupted by adding white zero-mean uniform noise. The signals length is 1000 samples. The mixture is computed as $\mathbf{x} = \mathbf{As}$, where **A** is defined as:

$$\mathbf{A}=\begin{bmatrix} 1.5-0.5i & -0.4+0.6i & 0.5+0.1i & 0.1 \\ 0.5+i & 0.4+0.2i & -0.1+0.3i & 0.2+0.5i \\ 0.2-0.4i & -0.6+i & 1.3+0.6i & 0.4+i \\ 0.2+i & -0.7+0.1i & -0.1-0.1i & 0.8-0.7 \end{bmatrix} \qquad (16)$$

The first row of figure 3 depicts the original independent signals while the second row shows the obtained four mixtures. The mixtures **x** have been firstly subtracted of their mean value and then processed (pre-whitened) by a 4×4 complex matrix **V** to make the covariance matrix of the signal $\mathbf{x}'=\mathbf{Vx}$ equal to the identity matrix. Signals $\mathbf{x}'$ was used to training the un-mixing matrix **W** and the spline's control points by equations (15). The learning rates *m* and *h* were chosen equal to 0.005 and halved every 6 passes (training epochs) through the complete time series. The selected adaptive spline structure had 30 control points sampled with Δu = 0.17, while the initial shape of the function was set to a hyperbolic tangent.

The last row in figure 3 shows network's outputs after the learning phase. It can be observed that the network is able to recover the original signals except for a phase shift, a change of scales and a random permutation. This fact is reflected in the permutation structure of the matrix **WA**:

$$\mathbf{WA}=\begin{bmatrix} 0.015+0.012i & 0.010+0.008i & \underline{0.200+0.417i} & 0.009+0.001 \\ -0.004i & \underline{0.608+0.012i} & 0.002+0.004i & 0.002+0.002i \\ \underline{0.500+0.010i} & 0.003-0.002i & 0.001+0.004i & 0.002+0.001i \\ 0.007-0.004i & 0.006-0.007i & 0.007+0.007i & \underline{-0.199-0.716i} \end{bmatrix}$$

Figure 5 shows the signal to noise ratio vs. the number of samples presented as input to the network, while figure 4 depicts the modulus of one complex activation function at the end of the learning phase.

4. CONCLUSIONS

In this paper a new adaptive non-linear function for blind complex domain signal processing has been proposed and applied to the problem of separating complex signals from instantaneous mixtures by maximizing the entropy of the outputs. It is based on a couple of spline functions, one for the real and one for the imaginary part of the input, whose control points are adaptively changed. A suitable adaptation algorithm based on information maximization has been derived and some experimental results are presented that demonstrate the effectiveness of the method.

The proposed approach allows to obtain good results with a low computational burden, since the complexity, both in forward and adaptive mode, is independent of the number of control parameters.

5. REFERENCES

[1] A.J. Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation*, Vol. 7, pp 1129-1159, 1995.

[2] E. Bingham, A. Hyvarinen,"ICA of complex valued signals: a fast robust deflationary algorithm", Proc. of IJCNN, Como, Italy, July, 2000.

[3] S. Fiori, A. Uncini, F. Piazza, "Neural Blind Separation of Complex Sources by Extended APEX Algorithm (EAPEX)", *Proc. of IEEE ISCAS'99*, Orlando (USA), June 1999, vol. V, pp. 627-630.

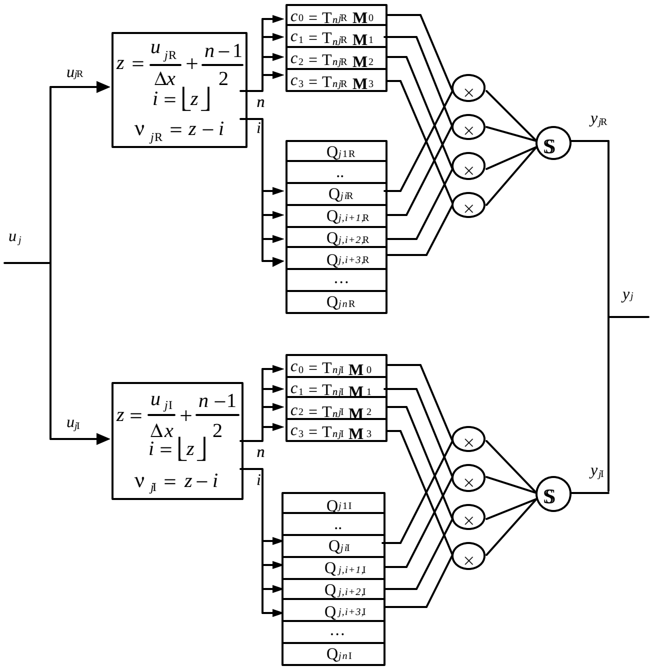
[4] L. Xu, C.C. Cheung, H.H. Yang, S. Amari, "Independent component analysis by the information theoretic approach with mixture of densities", *Proc. of IEEE ICNN-97*, Houston TX, USA, vol. 3, pp. 1821-1826, June 1997.

[5] A. Pierani, F. Piazza, M. Solazzi, A. Uncini, "Low Complexity Adaptive Non-Linear Function for Blind Signal Separation", *Proc. of IEEE IJCNN2000*, Como (Italy)

[6] S. Guarnieri, F. Piazza and A. Uncini, "Multilayer Feedforward Networks with Adaptive Spline Activation Function", *IEEE Trans. on Neural Network*, Vol. 10, No. 3, pp.672-683, May 1999

[7] A. Uncini, L. Vecci, P. Campolucci and F. Piazza, "Complex-valued Neural Networks with Adaptive Spline Activation Function for Digital Radio Links Nonlinear Equalization", IEEE Trans. on Signal Processing, Vol. 47, No. 2, February 1999.

[8] S.I. Amari, A. Cichocki, "Adaptive Blind Signal Processing - Neural Network Approaches", Proceedings of IEEE, Vol. 86, No. 10, Oct. 1998.



u_j
u_{jR}
$z=\frac{u_{jR}}{\Delta x}+\frac{n-1}{2}$
$i=\lfloor z\rfloor$
$\nu_{jR}=z-i$
n
i
$c_0 = T_{njR}\,\mathbf{M}_0$
$c_1 = T_{njR}\,\mathbf{M}_1$
$c_2 = T_{njR}\,\mathbf{M}_2$
$c_3 = T_{njR}\,\mathbf{M}_3$
Q_{j1R}
..
Q_{jiR}
$Q_{j,i+1,R}$
$Q_{j,i+2,R}$
$Q_{j,i+3,R}$
…
Q_{jnR}
×
S
y_{jR}
y_j
u_{jI}
$z=\frac{u_{jI}}{\Delta x}+\frac{n-1}{2}$
$i=\lfloor z\rfloor$
$\nu_{jI}=z-i$
n
i
$c_0 = T_{njI}\,\mathbf{M}_0$
$c_1 = T_{njI}\,\mathbf{M}_1$
$c_2 = T_{njI}\,\mathbf{M}_2$
$c_3 = T_{njI}\,\mathbf{M}_3$
Q_{j1I}
..
Q_{jiI}
$Q_{j,i+1,I}$
$Q_{j,i+2,I}$
$Q_{j,i+3,I}$
…
Q_{jnI}
×
S
y_{jI}

Figure 2 – The B-Spline architecture deriving from equations (2)-(5).

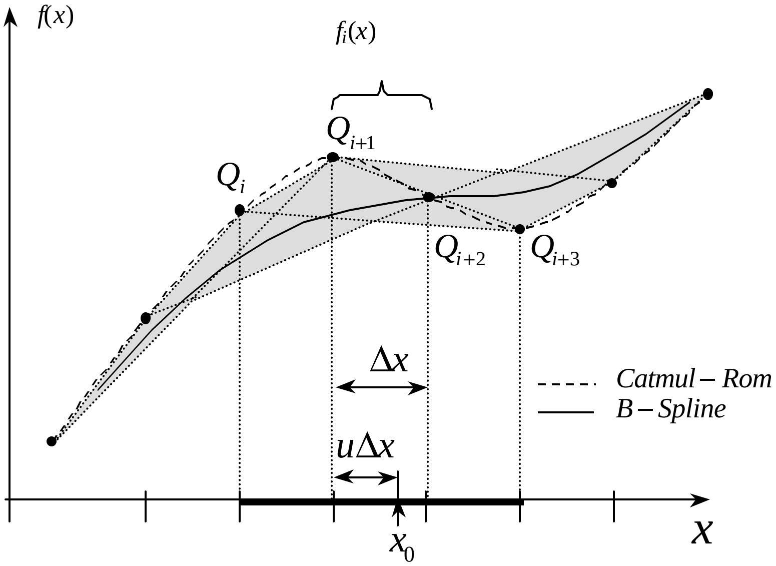


Figure 1 – The i-th B-Spline span with a fixed step Δx . $Q_i , Q_{i+1} , Q_{i+2} , Q_{i+3}$ represent its control points while u represents the local span variable.

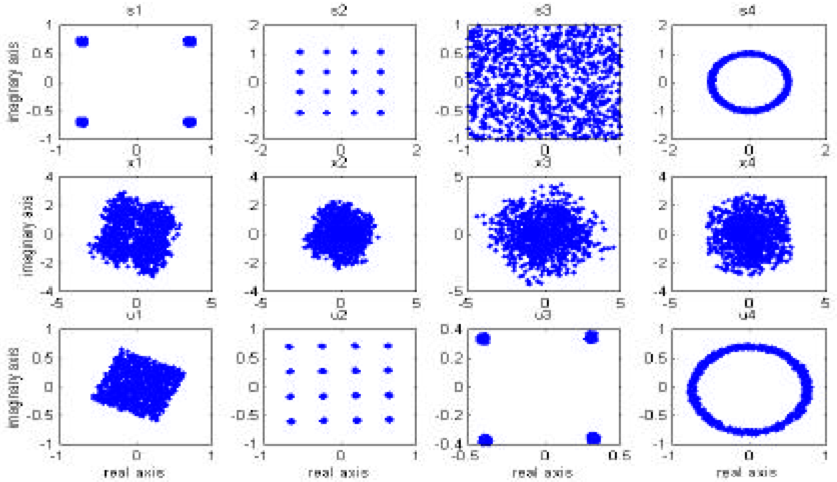


Figure 3 –
First row: s_1,..s_4, original signals;
second row: mixed signals;
third row: separated signal after 30 training epochs.

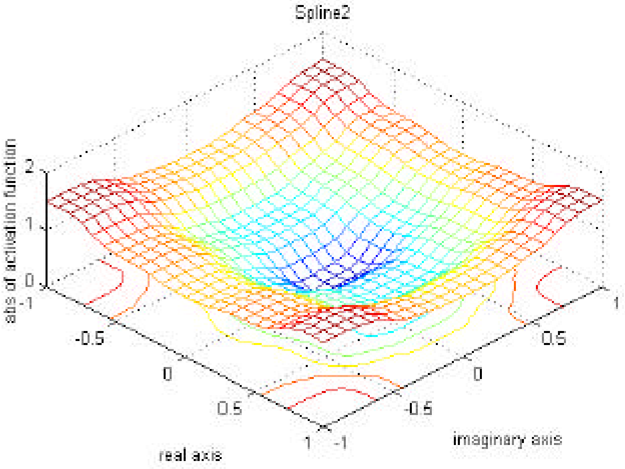


Figure 4 – Example of the activation function (absolute value) for the 16QAM separated signal.

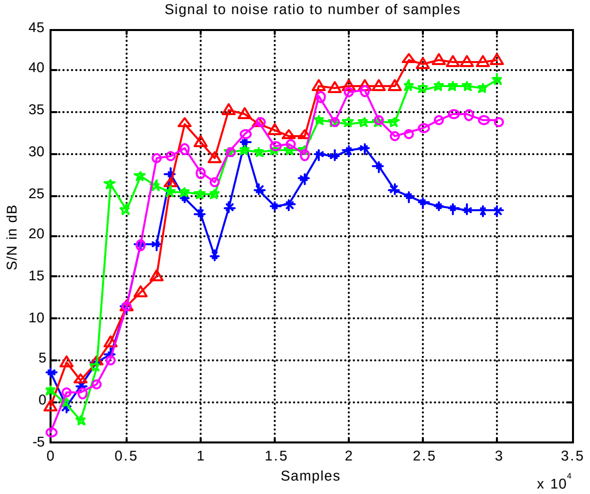


Figure 5 – Signal to noise ratio (SNR) of the separated signals vs. training epochs.