# Blind Deconvolution and Blind Source Separation
## *(A Summary)*

Shane M. Haas

February 8, 2001

## 1 Summary

The goal of blind deconvolution and source separation is to unravel the effects of an unknown linear transformation on a unknown signal source. For blind deconvolution, the transformation is a linear finite-impulse response (FIR) filter, and for blind source separation it is a matrix of mixing coefficients. A general architecture for these blind adaptive algorithms consists of an adjustable linear transformation and a non-linear function. In this presentation, we will examine the Bussgang algorithm for blind deconvolution [2] and an information-maximization algorithm to perform blind source separation [1].

## 2 Introduction

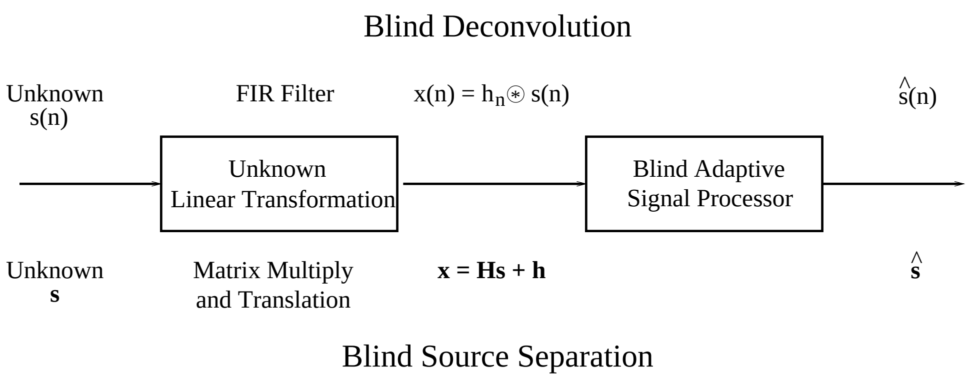


Figure 1: The blind deconvolution and blind source separation problem

Figure 1 illustrates the very similar problems of blind deconvolution and blind source separation. In blind deconvolution, the goal is to recover a signal after it has passed through an FIR filter. Similarly, the goal of blind source separation is to "demix" a vector of signals after they have passed through a matrix multiplication and translation operation. The problems are called "blind" because the signal processor only has access to the output of the linear transformation, and does not know the original source signal or the transformation.

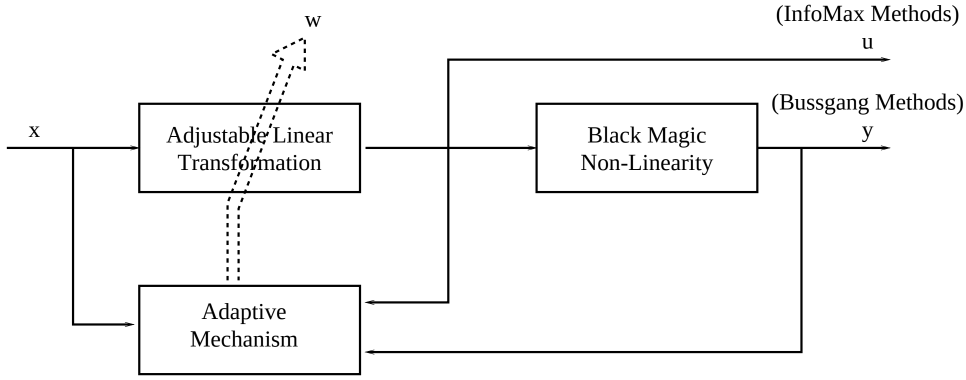


Figure 2: The general architecture of a blind adaptive algorithm

A general architecture for blind adaptive algorithms is shown in Figure 2. The signal passes through an adjustable linear transformation (FIR filter for blind deconvolution or matrix multiply and translation for blind source separation) and into a mysterious non-linear function. The adaptive mechanism adjusts the linear transformation in such a way as to optimize some objective function. Two pertinent questions arise:

- How to choose the non-linearity?
- What should the adaptive mechanism optimize?

Even if we assume that the architecture shown in Figure 2 is the correct one for solving these problems, neither of these questions have definitive answers. Blind source separation and blind deconvolution are active areas of research, and many algorithms that seem to work well are ad-hoc and dabble in the subtle art of voodoo.

We will examine two such algorithms in this presentation. The first method is the Bussgang algorithm for blind deconvolution [2]. It uses the celebrated Least-Mean Squares (LMS) algorithm as its adaptive mechanism and a Bayes estimator as its non-linear function. The second method, proposed by Bell and Sejnowski in [1], is sometimes called "independent component analysis." It uses an adaptive gradient algorithm to maximize the information content of the output of a non-linear "logistic" function. It is somewhat akin to "principle component analysis" but seeks to decompose a signal into statistically independent rather than just uncorrelated components. While this algorithm can be used for both blind deconvolution and blind source separation, we will only examine its application to the blind source separation problem.

# 3 The Bussgang Algorithm for Blind Deconvolution

We will first look at the problem of blind deconvolution using the Bussgang algorithm.

## 3.1 The Non-Linearity: Bayes Estimator

The Bussgang algorithm for blind deconvolution uses the Bayes estimator (conditional expectation) of the source given the equalized output as its non-linear function. Let $a_k$ denote the impulse response of a filter that would ideally equalize the unknown FIR filter $h_k$, i.e. $\sum_k a_k h_{n-k} = \delta_n$ where $\delta_n$ is the discrete unit impulse. We can then write the output of the adjustable FIR filter $w_k(n)$ at time $n$ as

$$\begin{aligned} u(n) &= \sum_k w_k(n) x(n-k) \\ &= \sum_k a_k x(n-k) + \sum_k \left(w_k(n) - a_k\right) x(n-k) \\ &= s(n) + v(n) \end{aligned}$$

where $s(n)$ is the original source signal and $v(n)$ is the convolutional noise defined as

$$v(n) = \sum_k (w_k(n) - a_k) x(n-k)$$

The Bussgang algorithm uses the following approximations. Assume that $v(n)$ is

- zero-mean,
- Gaussian,
- white, and is
- statistically independent of the source sequence $s(n)$.

This approximation is not that bad if $s(n)$ is a sequence of independent, zero-mean random variables, and the difference between the adjustable FIR filter $w_k(n)$ and the ideal equalizing filter $a_k$ is a long oscillatory sequence. If we also assume that $s(n)$ is a sequence of iid uniform random variables on the interval $(-\sqrt{3}, \sqrt{3})$, then the minimum mean squared error (Bayes) estimator of $s(n)$ given $u(n)$ is ([2], pg. 734)

$$\begin{aligned} y(n) &= E\left[s(n)|u(n)\right] \\ &= \frac{1}{c_0} u(n) + \frac{\sigma}{c_0} \frac{Z(u(n) + c_0/\sqrt{3}) - Z(u(n) - c_0/\sqrt{3})}{Q(u(n) - c_0/\sqrt{3}) - Q(u(n) + c_0/\sqrt{3})} \end{aligned}$$

where $\sigma^2$ is the variance of $v(n)$, $c_0 = \sqrt{1-\sigma^2}$ is a normalizing factor, and

$$\begin{aligned} Z(u) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \\ Q(u) &= \frac{1}{\sqrt{2\pi}} \int_u^\infty e^{-\frac{z^2}{2}} dz \end{aligned}$$

The reason that we assume the source is a sequence iid uniform random variables is to approximate the transmission of a real baseband $M$-ary pulse amplitude modulation (PAM) sequence with unit variance. To extend this result to complex baseband QAM, apply the estimator to the real and imaginary components separately ([2], pg. 741).

## 3.2 The Adaptive Mechanism: The LMS Algorithm

The Bussgang algorithm uses the popular LMS algorithm [2] to adjust the weights of an FIR filter. The LMS algorithm is a recursive, steepest descent approximation to the optimal Wiener filter. It tries to adjust the weights $w_k(n)$ to minimize the mean squared error between the filter output and a "desired" signal. In the Bussgang algorithm, this desired signal is the output of the Bayes estimator.

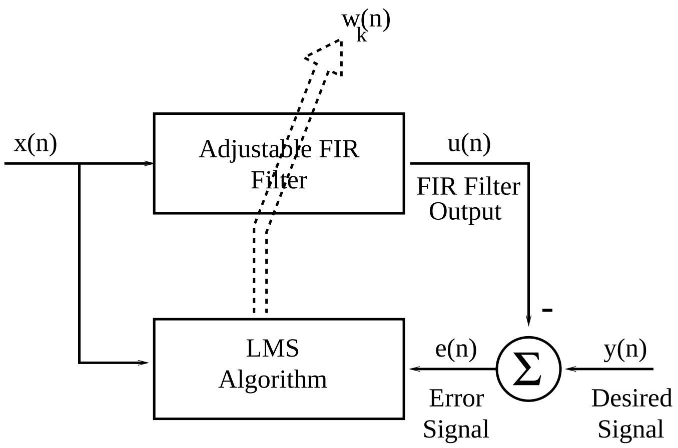


Figure 3: An adaptive filter using the LMS algorithm

Figure 3 shows the structure of an adaptive filter and the LMS algorithm. The LMS algorithm adjusts the weights $w_k(n)$ to minimize the mean squared error,

$$
\begin{aligned}
J(n) &= E\left[|e(n)|^2\right] \\
&= E\left[|y(n) - u(n)|^2\right] \\
&= E\left[|y(n) - \mathbf{w}^H(n)\mathbf{x}(n)|^2\right] \\
&= E[|y(n)|^2] - \mathbf{w}^H(n)\mathbf{p} - p^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{w}(n) &= (w_0(n), w_1(n), \ldots, w_{M-1}(n))^T \\
\mathbf{x}(n) &= (x(n), x(n-1), \ldots, x(n-M+1))^T \\
\mathbf{p} &= E[\mathbf{x}(n)y^*(n)] \\
\mathbf{R} &= E[\mathbf{x}(n)\mathbf{x}^H(n)]
\end{aligned}
$$

The principle of orthogonality states that the best M-tap FIR filter to minimize the mean squared error is the orthogonal projection of the desired signal onto the subspace generated by $\{x(n), x(n-1), \ldots, x(n-M+1)\}$. This optimal filter $\mathbf{w}$ makes the error signal $e(n)$ orthogonal to this subspace. This principle is shown graphically in Figure 4 and leads to the Wiener-Hopf condition of optimality:

$$
\begin{aligned}
E[\mathbf{x}(n)e^*(n)] &= E[\mathbf{x}(n)(y(n) - \mathbf{w}^H\mathbf{x}(n))^*] \\
&= \mathbf{p} - \mathbf{R}\mathbf{w} \\
&= 0
\end{aligned}
$$

In other words, the optimal FIR Wiener filter, $\mathbf{w}$, satisfies

$$\mathbf{R}\mathbf{w} = \mathbf{p}$$

This Wiener-Hopf equation can also be realized by setting the gradient of $J(n)$,

$$\frac{\partial J(n)}{\partial \mathbf{w}(n)} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)$$

equal to zero[1]. Using this gradient information, we can recursively adjust the weights $w_k(n)$ to minimize $J(n)$ in the following steepest-descent manner:

$$
\begin{aligned}
\Delta\mathbf{w} &= \mathbf{w}(n+1) - \mathbf{w}(n) \\
&\propto -\frac{\partial J(n)}{\partial \mathbf{w}(n)} \\
&\propto \mathbf{p} - \mathbf{R}\mathbf{w}(n)
\end{aligned}
$$

[1] We have assumed that $y(n)$ does not depend on $\mathbf{w}$; however, in the Bussgang algorithm, $y(n)$ is the output of the Bayes estimator acting on the output of the adjustable filter. As a result, the LMS algorithm in the Bussgang algorithm is actually trying to find a minimum of a non-convex cost function that has possibly more than one minimum.

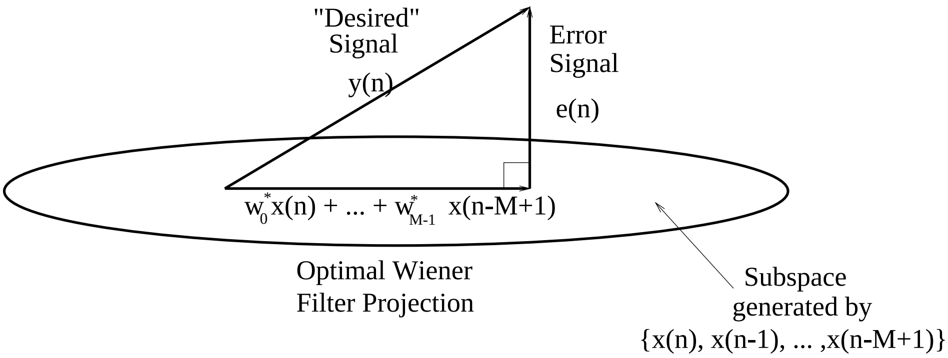


Figure 4: The optimal FIR filter in terms of mean squared error

If we use "instantaneous" values instead of the expected values, i.e. $\mathbf{p} \approx \mathbf{x}(n)y^*(n)$ and $\mathbf{R} \approx \mathbf{x}(n)\mathbf{x}^H(n)$, then we arrive at the LMS algorithm:

$$\begin{aligned} \Delta\mathbf{w} &\propto \mathbf{x}(n)y^*(n) - \mathbf{x}(n)\mathbf{x}^H(n)\mathbf{w}(n) \\ &= \mathbf{x}(n)(y(n) - \mathbf{w}^H(n)\mathbf{x}(n))^* \\ &= \mathbf{x}(n)e^*(n) \end{aligned}$$

It can be shown ([2], pg. 317) that the LMS algorithm converges in the mean to the optimal Wiener filter if the constant of proportionality is positive and less than twice the reciprocal of the maximum eigenvalue of the covariance matrix $\mathbf{R}$.

### 3.3 Summary of the Bussgang Algorithm

Figure 5 shows a block diagram of the Bussgang algorithm for blind deconvolution. Here:

**LMS Algorithm (For Complex Signals):**

$$\begin{aligned} \Delta\mathbf{w} &= \mathbf{x}(n)e^*(n) \\ \mathbf{w}(n) &= (w_0(n), w_1(n), \ldots, w_{M-1}(n))^T \\ \mathbf{x}(n) &= (x(n), x(n-1), \ldots, x(n-M+1))^T \\ e(n) &= y(n) - \mathbf{w}^H(n)\mathbf{x}(n) \end{aligned}$$

**Bayes Estimator (Apply to Real and Imaginary Components Separately):**

$$
\begin{aligned}
y(n) &= \frac{1}{c_0}u(n) + \frac{\sigma}{c_0}\frac{Z(u(n)+c_0/\sqrt{3}) - Z(u(n)-c_0/\sqrt{3})}{Q(u(n)-c_0/\sqrt{3}) - Q(u(n)+c_0/\sqrt{3})} \\
Z(u) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}} \\
Q(u) &= \frac{1}{\sqrt{2\pi}}\int_u^\infty e^{-\frac{z^2}{2}}dz \\
\sigma^2 &= var(v(n)) \\
c_0 &= \sqrt{1-\sigma^2}
\end{aligned}
$$

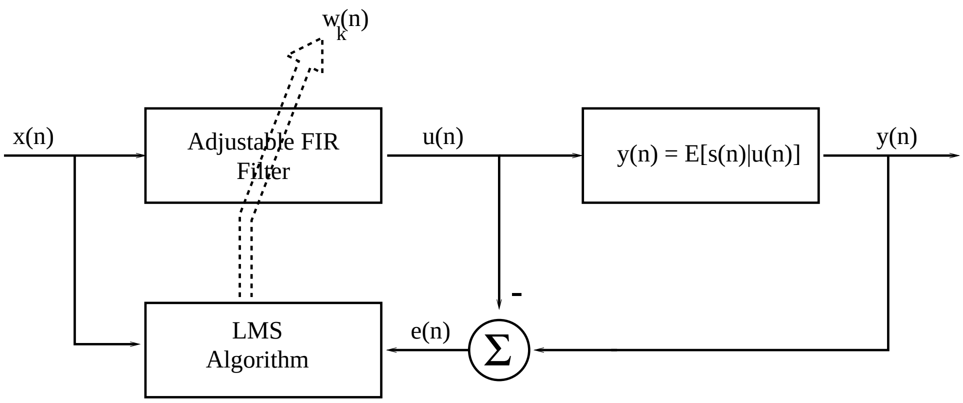


Figure 5: The Bussgang algorithm for blind deconvolution

# 4 An Information-Maximization Algorithm for Blind Source Separation

We will now examine the Bell and Sejnowski information-maximization approach to blind source separation. [1]

## 4.1 The Non-Linearity: The Logistic Function

As previously stated, choosing the non-linearity is somewhat of an art form. Some non-linearities work better than others on different problems. To perform blind source separation, Bell and Sejnowski use the "logistic" function

$$y_i = \frac{1}{1+e^{-u_i}}$$

where $u_i$ is the $i$th component of $\mathbf{u} = \mathbf{W}\mathbf{x}+\mathbf{w}$, $\mathbf{W}$ is an adjustable $N \times N$ matrix of demixing coefficients and $\mathbf{w}$ is an adjustable offset. All of these quantities are time-varying, but for notation simplicity we make this dependency implicit.

## 4.2 The Adaptive Mechanism: Information-Maximization

The Bell and Sejnowski algorithm adjusts the weights $\mathbf{W}$ and $\mathbf{w}$ to maximize the differential entropy of the non-linearity output. Assuming the non-linearity is an invertible function, the multivariate probability density function of its output, $\mathbf{y}$, is

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{f_{\mathbf{X}}(\mathbf{x})}{|J|}$$

where $J$ is the Jacobian

$$J = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial y_N}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_N} \end{vmatrix}$$

We can now write the differential entropy of $\mathbf{y}$ as

$$\begin{aligned} h(\mathbf{Y}) &= -E[\log f_{\mathbf{Y}}(\mathbf{y})] \\ &= E[\log |J|] - E[\log f_{\mathbf{X}}(\mathbf{x})] \end{aligned}$$

The second term is a differential entropy that does not depend on the adjustable weights $\mathbf{W}$ and $\mathbf{w}$. We will adjust the weights in a steepest ascent fashion using "instantaneous" values instead of expectations to maximize $h(\mathbf{Y})$:

$$\begin{aligned} \Delta\mathbf{W} &\propto \frac{\partial h(\mathbf{Y})}{\partial \mathbf{W}} \approx \frac{\partial \log |J|}{\partial \mathbf{W}} \\ \Delta\mathbf{w} &\propto \frac{\partial h(\mathbf{Y})}{\partial \mathbf{w}} \approx \frac{\partial \log |J|}{\partial \mathbf{w}} \end{aligned}$$

It is a rather tedious exercise in taking derivatives (see the appendix in [1] for details) but relatively straight forward to show that this adjustment rule simplifies to

$$\begin{aligned} \Delta\mathbf{W} &\propto [\mathbf{W}^T]^{-1} + (1 - 2\mathbf{y})\mathbf{x}^T \\ \Delta\mathbf{w} &\propto 1 - 2\mathbf{y} \end{aligned}$$

## 4.3 Summary of Information-Maximization Algorithm

Figure 6 shows a block diagram of an information-maximization algorithm for blind source separation. Here:

**InfoMax Algorithm (For Real Signals):**

$$\begin{aligned} \Delta\mathbf{W} &\propto [\mathbf{W}^T]^{-1} + (1 - 2\mathbf{y})\mathbf{x}^T \\ \Delta\mathbf{w} &\propto 1 - 2\mathbf{y} \end{aligned}$$

**Logistic Function (Apply Component-Wise):**

$$y_i = \frac{1}{1 + e^{-u_i}}$$

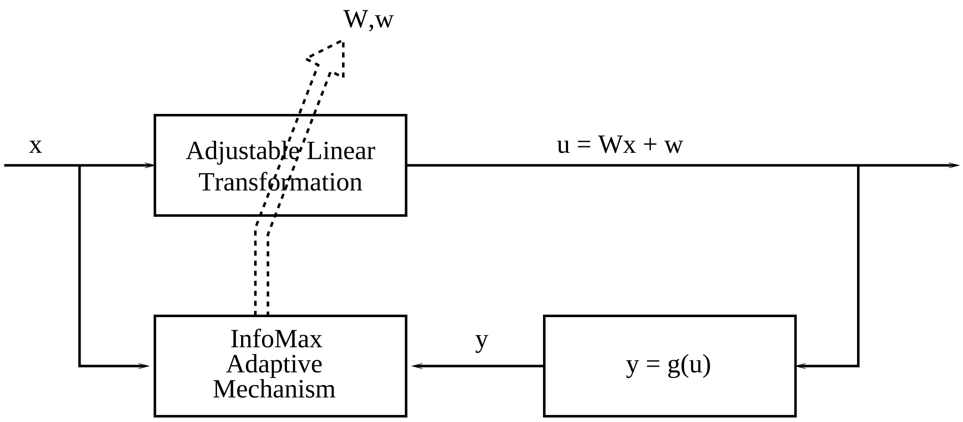


Figure 6: An information-maximization algorithm for blind source separation

# 5 Conclusions

We have examined the problems of blind deconvolution and blind source separation. For useful papers and nice demonstrations of these algorithms, see the following links:

- http://www.cnl.salk.edu/˜tony/ica.html
- http://sound.media.mit.edu/˜paris/ica.html
- http://sweat.cs.unm.edu/˜bap/demos.html

# References

[1] A.J. Bell and T.J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004–1034, 1995.

[2] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1991.