Blind Signal Separation

Ricky Der

Department of Electrical & Computer Engineering

McGill University

September 11, 2001 (Revised Nov. 2001)

Foreword

This report encapsulates work done as an intern at the Telecommunications and Signal Processing Laboratory of McGill University for the summer of 2001. My appreciation goes to Prof. Peter Kabal for providing me with an interesting topic, to the graduate students of TSP for their cute humor, and especially to Hossein Najafzadeh-Azghandi for his excellent technical advice.

Abstract

Blind Signal Separation is the task of separating signals when only their mixtures are observed. Recently, Independent Component Analysis has become a favourite method of researchers for attacking this problem. We review the techniques, from cumulant-based algorithms to Infomax to second-order statistics, from feedback to feedforward architectures, from the instantaneous to the convolutional problem. A new method for reducing the whitening effect on speech, known to occur in feedforward architectures, is introduced. The procedure also possesses significant stabilization properties, being based on performing the filter update in the LP-residual domain of speech. Experimental tests are conducted, and the algorithms compared.

Table of Contents

1 Introduction

Blind Signal Separation is the general problem of determining original sources when only their mixtures are available for observation. Over the past 5 years, research on this topic has exploded due to the emergence of relatively successful separation algorithms, as well as the growing sentiment that the technique constitutes a universal panacea capable of everything from de-noising speech to uncovering the laws of the stock market.

The process is often termed “blind”, with the understanding that both source signals and mixing procedure are unknown [1, 2]. Such a statement is of course blatant exaggeration – indeed the assumption of *some* specific mixing model is the paramount piece of prior information required, and in many scenarios even knowledge of certain source statistics is necessary. We thus begin with the channel model:

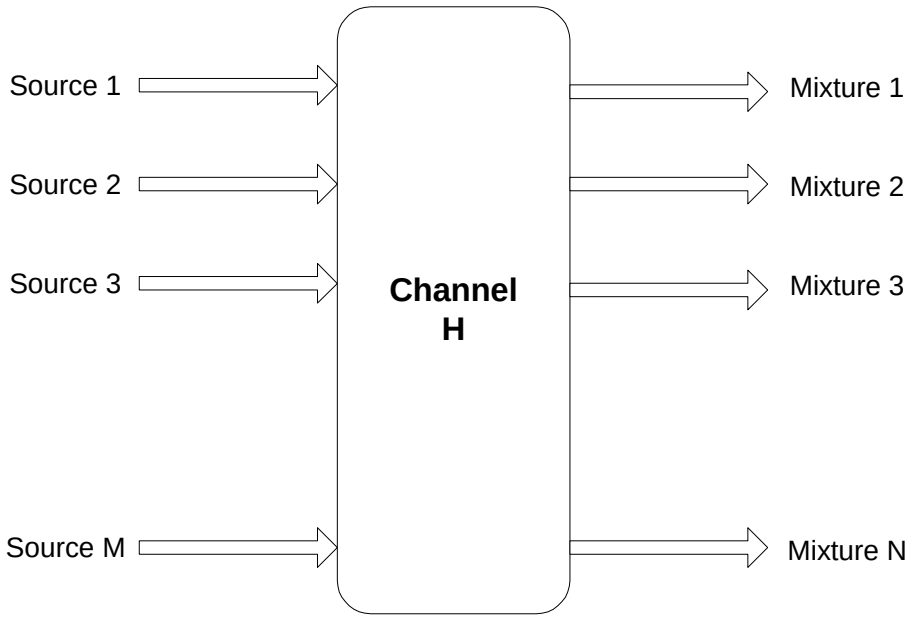


Figure 1: Block Diagram of the Mixing Model

The sources may be sounds, images, biomedical or financial data. Our primary interest will be in audio source signals, with microphones to collect the output mixed signals.

Under this setting, the channel **H** may generally be construed as a linear time-invariant (LTI) system, though there is some activity occurring with nonlinear mixing models (see for instance [1, 3]).

Three levels of complexity are discerned:

- **H** is a matrix. We call this the *instantaneous mixing model*, since only the relative attenuations of sound due to the microphone-source distances are accommodated.

- $H_{ij} = a_{ij} z^{-D_{ij}}$. This is the *delayed mixing model*, incorporating not only the attenuation a_{ij} between the i^{th} microphone and j^{th} source, but the travel time d_{ij} as well.

- A matrix of FIR filters $H_{ij} = \sum_{k=0}^{L-1} a_{ijk} z^{-k}$. This is the *convolutive mixing model*, where room reverberation is accounted for.

Further generalization admits a non-dimension preserving **H:** $N \neq M$ **.** Another attempt at realism introduces a dynamic environment equipped with moving speakers: $\mathbf{H} = \mathbf{H}(t)$. Finally, we may include microphone (sensor) noise $\mathbf{n}(t)$ with the model, though it is possible to consider noise as an additional source. For the latter reason we do not deal with noise in this report; however, methods are available (see [2, 4, 5]) for estimating and eliminating such deteriorating effects without analyzing them as sources.

Though many papers purport to introduce “new” methods of solution, the existing framework (and solutions) for blind signal separation are often the same. Sources are mod-

eled as random processes despite their essential deterministic mode of production[1], and the statistical independence of these random sources is then exploited[2]. Specifically, the determining criterion for separation is a measure of independence, typically represented by some cost function J . The extremum of J, with respect to the parameters of some inverse mixing process, then corresponds with more or less independent outputs. Such a system is illustrated in Fig 2:

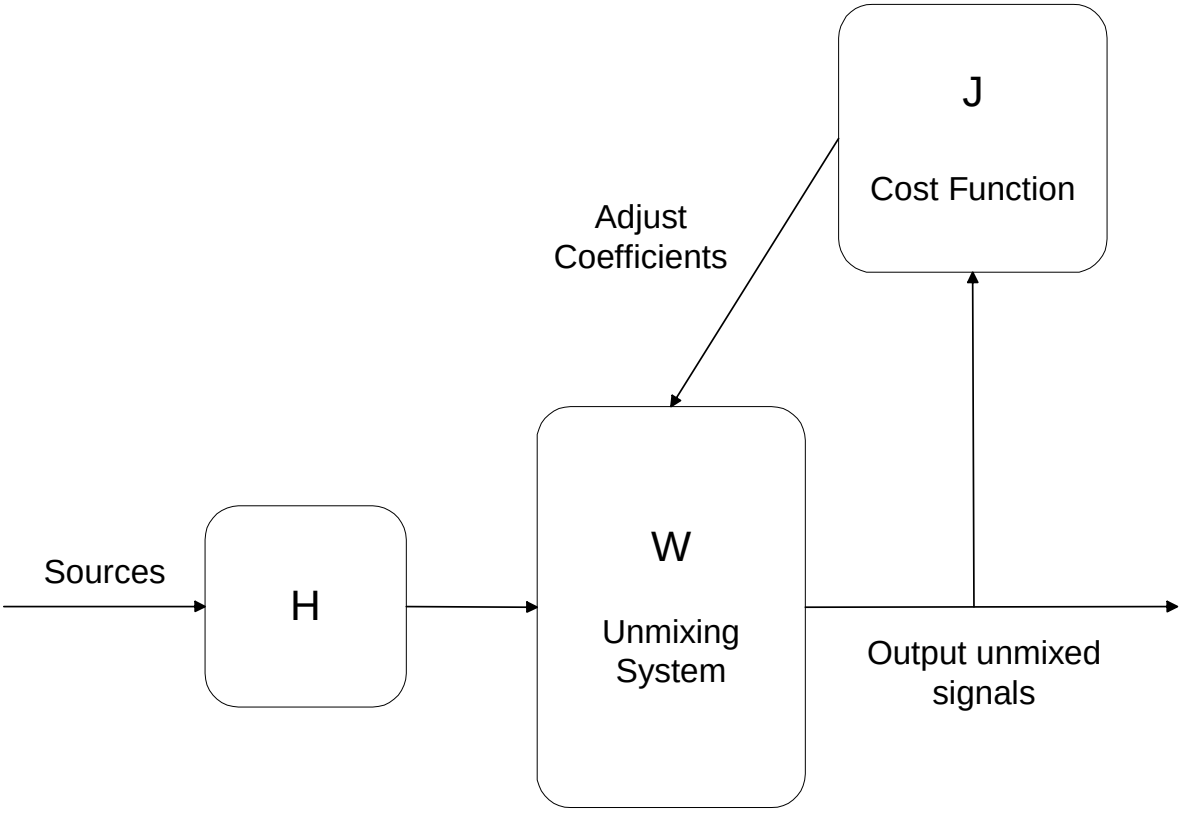


Figure 2: Block Diagram of a Separating System

[1] Justification for such procedures may be found in examining the relationships between maximal description length, Kolmogorov complexity and entropy, see [6].

[2] For an interesting algorithm in which statistical independence of sources seems not to be required see [7].

Algorithms which rely on this concept, the separation-independence equivalence, may be classed as those performing *Independent Component Analysis*. The problem of blind signal separation is then reduced to a mathematical *optimization* problem, upon which a multitude of tested techniques may be brought to bear.

The principal differences rest on the varieties of cost functions utilized. Researcher A swears by kurtosis, B by mutual information, C by cross power-spectra, D by negentropy and E by log-likelihood. In many cases these approaches are the result of superficially different formalisms, and can be shown to be mathematically equivalent [1]. Where real divergence remains, a particular path may be chosen on a case-by-case basis, depending on the requirements of the application (computational ease, source characteristics, stability etc.)

The purpose of this report is to provide an exposition of these procedures and to perform comparative experimental tests of the algorithms. Chapter II details the mathematical theory behind a number of approaches. Section 1 deals with instantaneous mixtures, Section 2 with delayed mixtures and Section 3 with convolved mixtures. Section 4 overviews a variety of optimization techniques, prime among them the natural gradient, to vastly increase convergence rate. Section 5 delineates our own contribution to the subject: an analysis of how LP filtering and an LP domain weight update may be used to improve the stability and convergence of feedforward adaptive deconvolution, in addition to reducing the well-known whitening effect.

Chapter III is concerned with experimental results, obtained through simulation, roughly paralleling the layout of Chapter II. The report ends with a conclusion of our work and an appendix containing a timeline of topics.

2 Theoretical Considerations

2.1 Instantaneous Mixtures

Let $\mathbf{s}(t)$ be a vector of *n* independent source random processes, and $\mathbf{x}(t)$ be a vector of *n* mixtures[3], obtained via:

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t), \qquad \mathbf{H} \in \mathbb{R}^{nxn}$$

Perfectly recovered outputs $\mathbf{u}(t) = \mathbf{W}\mathbf{x}(t)$ result if $\mathbf{WH} = \mathbf{I}$. Since the independence of sources is invariant to any amplitude multiplication, as well as to re-ordering within the vector, we may only expect $\mathbf{WH} = \mathbf{p}$, $\mathbf{p}$ some permuted and row-scaled version of the identity matrix. This is the well-known indeterminacy of recovery.

Before continuing it is apposite to discuss a point which is sometimes mysteriously neglected in the literature: the non-stationary/stationary qualities of the sources. It should be clear that within our framework we always speak of random processes, and not random variables. Yet frequently one sees reference to the static "kurtosis", or "probability density distribution", or the "entropy" of a source as if random processes were simply a series of i.i.d. random variables. Typically, all these statistics change with time, in a possibly arbitrary way.

Speech is often said to be "non-stationary" on the inter-frame level (due to the dynamic shape of the vocal tract [8]) and locally "quasi-stationary" within a frame (5-20 ms intervals). Yet global statements such as "A histogram of typical speech amplitudes approaches a gamma or Laplacian probability distribution" [9], abound. Clearly, the connotation behind these disparate attitudes is that stochastic attributes qualitatively differ depending upon the optic used. It is thus helpful to describe a three-stage taxonomy:

---

[3] We restrict ourselves to the square case, though greater robustness in the presence of noise can be obtained with more microphones than sources see [33].

- On the local (intra-frame) level of duration 5-20 ms, speech may be considered to be a stationary random process.

- On a frame-to-frame level, speech is non-stationary with dynamic auto-correlation function, local probability distribution etc. (e.g. the voiced parts possess a gamma/Laplacian distribution, whereas the unvoiced parts are more Gaussian.)

- On the global interval of hundreds of frames (seconds of speech), speech is a random variable. Static ensemble statistics may be discerned (e.g. speech histograms approach some fixed Laplacian distribution, with a fixed variance, kurtosis, entropy etc.).

In addition to these ideas, some assumption of ergodicity is required [10] in order to use time averages instead of ensemble expectations and to justify the convergence of local averages to global statistics.

2.1.1 Moment and Cumulant-Based Separation

Reiterating, the crux of the Independent Component Analysis solution rests on optimizing a cost function reflecting the independence of the outputs, with respect to channel coefficients. Classical “Principal Component Analysis”, “Karhunen-Loève Transform”, or Singular Value Decomposition” are specializations of the technique in that they require components only to be decorrelated (linearly independent). This rests upon diagonalization of the cross-covariance matrix.

For analytic probability density functions, a necessary and sufficient condition engendering statistical independence is that all higher-order cross-moments also diagonalize. Thus we require (assuming zero-mean sources):

$$E(X_i(t_1)X_j(t_2)X_k(t_3)...X_n(t_N)) = \delta_{ijk...N}, \quad N = 1,2,3...,$$

for all choices of times $t_1, t_2, ..., t_N$

Thankfully, full mathematical independence is unnecessary for auditory separation. A first simplification foregoes the need to diagonalize at different times, requiring only instantaneous diagonalization:

$$E(X_i(t)X_j(t)X_k(t)...X_N(t)) = F(t) = \delta_{ijk...N},$$

However, such an objective still requires considerable computational power, since the number of components in the *N*-th moment tensor increases exponentially with *N*. Moreover, the time-average estimation of higher-order statistics necessitates much larger sample sizes than their second-order counterparts [11]. This estimation non-robustness can cause problems with statistical series not truncated at just 4^{th} order, leading to excessive fluctuations at the tail-ends of the distribution [1].

Instead of moments, it is often convenient to use a different measure: the cumulants of a random process, which possess special physical significance for Gaussian-like distributions. In particular, the first three cumulants c_k of a random variable are precisely the first three (central) moments:

$$c_1 = \mu$$
$$c_2 = \sigma^2$$
$$c_3 = E[X^3]$$

For symmetric distributions, c_3 is zero, so one must move a step further:

$$c_4 = E[X^4] - 3\sigma^2$$

The fourth cumulant is then the first relevant statistic higher than second order. Normalized by variance, we obtain an energy-invariant characterization of amplitude spread:

$$kurtosis = \kappa \equiv \frac{c_4}{\sigma^2} = \frac{E[X^4]}{E[X^2]} - 3$$

Suppose we wish to diagonalize the *N*-th order cross-cumulant tensor T_N. A variety of cost functions J are possible[4]; a particularly useful one is the sum of squares of the diagonal components under a rotation, which maximizes when T_N is diagonal:

$$J = [c_n(u_1)]^2 + [c_n(u_2)]^2 + ... + [c_n(u_N)]^2$$

where $\mathbf{u} = \mathbf{W}\mathbf{x}$

The extremum is simply found via the gradient:

$$\frac{\partial J(\mathbf{W}_{\mathbf{opt}})}{\partial \mathbf{W}} = 0$$

This equation, though in general nonlinear, can be iteratively solved, for instance, by any number of optimization routines (such as steepest ascent). Alternatively, it is possible to use a Singular Value-like decomposition to diagonalize directly. Such approaches are essentially the one taken in the JADE algorithm [12], by Cardoso, and the contrast function of Comon [13] specialized to the fourth cumulant[5].

Many cumulant-based algorithms require no adaptation, as iterative techniques are not necessary since the diagonalization is often a purely algebraic problem. Still, being an explicit processing of higher-order statistics, they are difficult and non-robust, and sensitive to outliers [10]. Moreover, the estimation of the expectations involves a global, batch-based method requiring intensive computation [1]. Use of only second-order statistics reduces the

[4] A list of criteria cumulant-based cost functions ought to satisfy is given in [15].

[5] See [14], and the generalization to complex-valued signals and higher-orders in [15].

level of complexity, but in the absence of cross-time-lag information is often inadequate for separation. We thus investigate other options.

2.1.2 Information Maximization

In 1995, Bell & Sejnowski [16] introduced a cost function taking into account, in an implicit fashion, *all* higher-order statistics. Their primary novelty was to use Mutual Information, an information-theoretic quantity, as the measure of statistical independence. We provide a brief account of their motivation:

The Mutual Information $I(X,Y)$ is a quantity introduced by Shannon which quantitates the degree of overlap between two random variables, *X* and *Y*. It is always positive, and zero if and only if X and Y are independent [6]. Writing the mutual information between *n* variables in terms of a more measurable quantity, the joint entropy $H(y_1, y_2, ..., y_n)$, is possible via the chain rule:

$$H(y_1,...,y_n) = H(y_1) + ... + H(y_n) - I(y_1,...,y_n), \quad \text{which is by definition}$$

$$H(y_1,...,y_n) = -E[\log p(y_1)] - ... - E[\log p(y_n)] - I(y_1,...,y_n)$$

where *p* is the probability density function of each variable y_i.

The above equation demonstrates clearly that simply maximizing the joint entropy of the outputs **y=u** is *not* the same as minimizing the mutual information, due to the interfering marginal entropy terms. However, if $\mathbf{y} = g(\mathbf{u})$, where *g* is an invertible function so that $p(y_i) = \frac{p(u_i)}{\left|\frac{\partial y_i}{\partial u_i}\right|}$ (by simple Jacobian transformation), then the marginal terms can be eliminated by setting $\frac{\partial y_i}{\partial u_i} = g'(u_i) = p(u_i)$.

In this case we have

$$H(y_1,...,y_n) = -I(y_1,...,y_n).$$

Thus maximization of the joint entropy of **y** is equivalent to minimizing the mutual information between the components of **y**. This in turn renders the outputs **y** independent, and, by invertibility of the function *g*, also the outputs **u**.

A detailed discussion of the nonlinearity *g* is postponed until Section 2.4.3. In the meantime, assuming an appropriate *g* may be found so that the marginal error terms are negligible, we have obtained an information-theoretic cost function:

$$J = H(\mathbf{y}) = H(g(\mathbf{u})) = -E[\log p(g(\mathbf{W}\mathbf{x}))]$$

From here, $\frac{\partial J}{\partial \mathbf{W}}$ gives a *deterministic* gradient ascent direction by which to determine the maximum. Due to the expectation operator, however, this involves block estimation of averages over **x**. An alternative attack is to remove the expectation operator, thus using *stochastic* gradient. This gradient is perturbed by the local random motion of **x**, but still eventually converges given the averaging effect on search directions on a global scale (Figure 3). Stochastic gradient methods enjoy the special advantage of tracking capability.

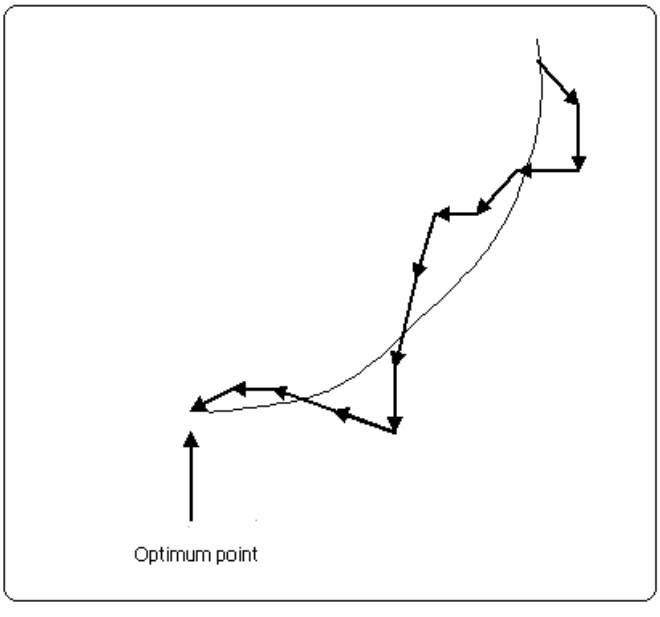


Figure 3: An illustration of the locus of a deterministic gradient (solid line), and stochastic gradient (arrowed line)

Our final objective function is thus:

$$J = \log\{p(g(\mathbf{Wx}))\}$$

The computation of $\frac{\partial J}{\partial \mathbf{W}}$ is a straightforward exercise in matrix calculus; interested readers may consult [1] for a derivation. We produce here the ultimate result:

$$\Delta \mathbf{W} \propto \frac{\partial J}{\partial \mathbf{W}} = (\mathbf{W}^T)^{-1} + \frac{g''(\mathbf{u})}{g'(\mathbf{u})}\mathbf{x}^T$$

A gradient ascent update is then given by:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \mu\left[(\mathbf{W}^T(t))^{-1} + \frac{g''(\mathbf{u})}{g'(\mathbf{u})}\mathbf{x}^T\right]$$

where μ is the step-size or learning rate. Following convention, it is useful to define the nonlinearity $\varphi(\mathbf{u}) = \frac{g''(\mathbf{u})}{g'(\mathbf{u})}$ (termed the *score* function):

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \mu\left[(\mathbf{W}^T(t))^{-1} + \varphi(\mathbf{u})\mathbf{x}^T\right] \qquad \text{(Eq. 1)}$$

A similar approach, based on like-minded intuition, is to maximize the entropy of the outputs **u** relative to the Gaussian distribution. This relative entropy is called *negentropy*, and is defined by [6]:

$$J(\mathbf{u}) = D(p(\mathbf{u}) \,\|\, p_G(\mathbf{u})) = \int_{\mathbb{R}^N} p(\mathbf{u}) \log \frac{p(\mathbf{u})}{p_G(\mathbf{u})} d\mathbf{u}$$

A special case of the Kullback-Leibler divergence, it is something of a metric measuring the statistical distance between a given distribution $p(\mathbf{u})$ and a Gaussian distribution with the same mean and variance as $p(\mathbf{u})$. Lee has shown in [1] that the maximization of the joint entropy of **y** is mathematically equivalent to the maximization of relative entropy of **u**; this

gives some physical insight into information maximization: in effect we drive the output signals as far away from Gaussian as possible.

2.1.3 Natural Gradient Algorithm

A much more efficient search direction manifests by post-multiplying the entropy gradient in equation (1) by $\mathbf{W}^T\mathbf{W}$ [17]:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \mu\left[(\mathbf{W}^T(t))^{-1} + \varphi(\mathbf{u})\mathbf{x}^T\right]\mathbf{W}^T(t)\cdot\mathbf{W}(t)$$

$$= \mathbf{W}(t) + \mu\left[\mathbf{I} + \varphi(\mathbf{u})\mathbf{u}^T\right]\mathbf{W}(t) \qquad \text{(Eq. 2)}$$

This results in the so-called *natural gradient* algorithm. The reader should be aware that the natural gradient is not simply a stumbled-upon empirical heuristic, but a very general mathematical optimization tool entertaining wide applications. A full explication is given in Section 2.4.1, but a simple intuition can be obtained by noticing the standard gradient in Eq. 1 has different units on either side, hence a convergence depending on the axis scaling [1], whereas the natural gradient algorithm normalizes by **W**, rendering the gradient invariant to such scaling. Surprisingly, this enhancement comes at *lower* computational cost, removing the need to perform a matrix inversion (compare Eq. 2 with Eq. 1).

2.1.4 Separation Based on Second-order Statistics

We alluded in Section 2.1.1 that second-order statistics are not usually sufficient for the separation of sources. The reasoning can be formulated mathematically, following [18]:

Assume an instantaneous mixture model of:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \qquad \mathbf{A} \in \mathbb{R}^{nxn}$$

An intra-frame measurement of the mixed cross-covariance matrix produces:

$$\mathbf{R} = E(\mathbf{x}(t)\cdot\mathbf{x}^T(t)) = \mathbf{A}\left\langle\mathbf{s}(t)\cdot\mathbf{s}^T(t)\right\rangle\mathbf{A}^T$$

Due to the symmetry of **R,** this gives only $\frac{n(n+1)}{2}$ equations for n^2 coefficients of **A**, and n auto-covariances in $\langle \mathbf{s}(t) \cdot \mathbf{s}^T(t) \rangle$ (recall the sources are mutually independent). Even constraining the scaling indeterminacy by setting $A_{ii} = 1$, i=1,...,n, there remain $\frac{n(n+1)}{2}$ equations for n^2 unknowns.

The above demonstrates the poverty of second-order statistics. *However*, one can use the non-i.i.d. character of most random processes to advantage. Suppose that a given random process is wide-sense stationary. Then the cross-covariance matrix can also be measured at time lags, with differing results:

$$\mathbf{R}(\tau) = E(\mathbf{x}(t) \cdot \mathbf{x}^T(t+\tau)) = \mathbf{A} \langle \mathbf{s}(t) \cdot \mathbf{s}^T(t+\tau) \rangle \mathbf{A}^T$$

A measurement of $\mathbf{R}(0)$ *and* $\mathbf{R}(\tau)$ for some τ gives $n(n+1)$ independent equations for $n^2 - n$ unknown channel coefficients and $2n$ unknown source covariances: a well-determined system. The solutions may be found within a non-symmetric eigenvalue problem as described in [18].

Even more interestingly, if the random process is *non-stationary*, instantaneous measurements of the cross-covariance at various times will give independent equations without resorting to time-lags [4]:

$$\mathbf{R}(t) = E(\mathbf{x}(t) \cdot \mathbf{x}^T(t)) = \mathbf{A} \langle \mathbf{s}(t) \cdot \mathbf{s}^T(t) \rangle \mathbf{A}^T$$

In particular, K measurements of the cross-covariance matrix **R** at times $t_{1,\ldots,} t_k$ produces $\frac{Kn(n+1)}{2}$ equations in $n^2 + Kn - n$ unknowns. Sufficient conditions for well-determinedness are possible as long there are $K \geq 2$ measurements.

A peculiar landsdcape thus arises: the *richer* the source characteristics, the *greater* the separation capability of second-order statistics. This is analogous to the fact that a mixture of

Gaussians is the most difficult to separate *just because* they are completely characterized by only two statistical parameters: mean and variance.

A cost function can be obtained by interpreting the K instantaneous cross-covariance measurements $\hat{\mathbf{R}}(k)$ as imperfect estimates of an underlying, dynamic $\mathbf{R}(k)$ [4]. First define an error term:

$$\mathbf{E}(k) = \hat{\mathbf{R}}(k) - \mathbf{R}(k) = \hat{\mathbf{R}}(k) - \mathbf{A}\left\langle \mathbf{s}(k) \cdot \mathbf{s}^T(k) \right\rangle \mathbf{A}^T = \hat{\mathbf{R}}(k) - \mathbf{A}\Lambda(k) \cdot \mathbf{A}^T,$$

and a total error $J = \sum_{k=1}^{K} \| \mathbf{E}(k) \|^2$, where the norm is the sum of squares of matrix elements: $\| \mathbf{E}(k) \|^2 = \sum_i \sum_j E_{ij}{}^2 = trace(\mathbf{E}\mathbf{E}^T)$.

It is then reasonable to estimate the values of the mixing matrix **A** and source covariances $\Lambda(k)$ by minimizing the squared error with respect to these unknowns:

$$\hat{\mathbf{A}}, \hat{\Lambda}(k) = \underset{\mathbf{A}, \Lambda(k), A_{ii}=1}{\arg\min} \sum_{k=1}^{K} \| \mathbf{E}(k) \|^2$$

The extremum may be found be computing gradients:

$$\frac{\partial J}{\partial \mathbf{A}} = -4 \sum_{k=1}^{K} \mathbf{E}(k) \cdot \mathbf{A}\Lambda(k) = \mathbf{0}, \qquad \frac{\partial J}{\partial \Lambda(k)} = -2 * diag(\mathbf{A}\mathbf{E}(k) \cdot \mathbf{A}^T) = \mathbf{0}$$

Note that the diagonalization function ($diag(\mathrm{B})_{ij} = \mathrm{B}_{ij}\delta_{ij}$) makes use of our *a priori* knowledge of source independence and further constrains the gradient. The second equation can be solved easily:

$$\Lambda(k) = diag(\mathbf{A}^{-1}\hat{\mathbf{R}}(k) \cdot (\mathbf{A}^T)^{-1}) \qquad \text{(Eq. 3)}$$

The first equation, however, is more complicated and probably the simplest method of obtaining the minimum is via (deterministic) gradient descent:

$$\mathbf{A}(t+1) = \mathbf{A}(t) - \mu \frac{\partial J}{\partial \mathbf{A}} = \mathbf{A}(t) + \mu \sum_{k=1}^{K} \mathbf{E}(k) \cdot \mathbf{A}(t) \cdot \Lambda(k) \qquad \text{(Eq. 4)}$$

Equations 3 and 4 are coupled; a continuous iteration between the two formulas provides an instantaneous separating algorithm based on only second-order statistics. For the case of speech, this algorithm is not local, unlike stochastic gradient, since it is necessary to choose the *K* covariance estimates far enough apart to render independent equations.

2.2 Delayed Mixtures

Adopting the same source assumptions as in Section 2.1, the mixing process becomes:

$$x_i(t) = \sum_{j=1}^{n} a_{ij} z^{-D_{ij}} s_j(t), \qquad i = 1,...,n \qquad \text{(Eq. 5)}$$

A simplification is afforded, in analogy to the arbitrary scaling indeterminacy of Section 2.1, by realising that only *relative* delay is significant. Thus with no loss of generality[6] we consider exclusively cross-delays: $D_{ii} = 0$. This constraint also removes the delay ambiguity in the separated solution.

One may ask why the pure delay case should be studied given that the delay operator z^{-D} is a special case of general convolutional mixing. The answer is it may be useful for more general FIR filtering. At a sampling rate of 16 kHz, a meter of distance in air corresponds to a delay of 50 taps. These are 50 additional filter coefficients which must be appended to *each* deconvolving filter: a total of $50n^2$ additional coefficients to adapt. Thus it seems pertinent to study a scheme for adaptive delays in the hope that redundancy may be removed in the later problem.

The separation of delayed mixtures is not much more difficult than for that of instantaneous mixtures. The principal quandary rests in the selection of an un-mixing process. While a feedforward design is possible, we emulate [19] in the use of the following feedback network, here displayed in the 2x2 case:

---

[6] Nearly no loss of generality. It should be noted that this assumption of relative delay provides constraints on the nature of the channel assignment: i.e. that the "direct" source-microphone channels should correspond with the shortest paths so as to eliminate "negative delays" from equation descriptions.

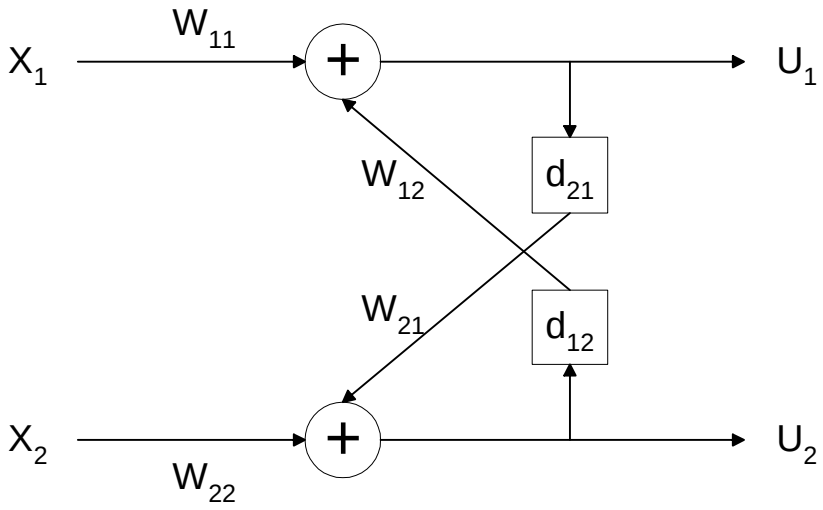


Figure 4: A feedback separation network with adaptive weights and delays

This network computes the output:

$$u_i(t) = w_{ii}x_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ij}u_j(t - d_{ij}) \qquad \text{(Eq. 6)}$$

From Eqs. 5 and 6 it is not hard to see that separation will occur if the adaptive delays d_{ij} converge to the mixing delays D_{ij}, $w_{ii} = w_{jj}$ for all i, j, and $w_{ij} = -a_{ij}$, for $i \neq j$. Essentially the direct source component of x_i is used to subtract the indirect component from x_j. The direct forward weights w_{ii} manipulate the overall scaling factor.

Again, by choosing a cost function *J*, a stochastic gradient algorithm can be derived. Once more, the information-theoretic objective $J = \log\{p(g(\mathbf{u}))\}$ will be considered. The derivations for the gradients $\frac{\partial J}{\partial w_{ij}}, \frac{\partial J}{\partial d_{ij}}$ are trivial[7]; we quote the final result:

$$\Delta w_{ii} \propto \frac{g''(u_i)}{g'(u_i)} x_i + \frac{1}{w_{ii}}, \text{ for all } i \qquad \text{(Eq 7)}$$

$$\Delta w_{ij} \propto \frac{g''(u_i)}{g'(u_i)} u_j(t - d_{ij}), \text{ for all } i \neq j \qquad \text{(Eq 8)}$$

$$\Delta d_{ij} \propto \frac{-g''(u_i)}{g'(u_i)} w_{ij} \frac{d}{dt}(u_j(t - d_{ij})) \qquad \text{(Eq 9)}$$

These 3 equations define the standard gradient feedback Infomax algorithm for the separation of delayed sources.

[7] There is a slight difficulty when taking the delay derivative however, since the outputs in the feedback system are functions of not only the current time but previous time values as well. Backwardly-expanded time derivatives are needed to fully account for all contributions; however, this added information may not aid significantly and indeed may hinder convergence [19]. Here we consider only the simpler gradient rule.

2.3 Convolved Mixtures

Here, the mixing process assumes the form:

$$x_i(t) = \sum_{j=1}^{n} \sum_{k=0}^{L-1} a_{ijk} s_j(t-k) \qquad \text{(Eq. 10)}$$

The assumption of a full matrix of mixing FIR filters $H_{ij} = \sum_{k=0}^{L-1} a_{ijk} z^{-k}$ models the most realistic scenario for acoustical signals. As with delay, the solution to the convolutional problem is in principle no more difficult than the instantaneous case: an inverse filtering system separates, adapting the filter coefficients by optimizing some cost function. In practice however, it is a far more formidable procedure since the interference from other sources must be cancelled not at a single lag, but at all lags up to the filter length L. This can be large: thousands of taps.

Moreover, the scaling indeterminacy of the outputs is exacerbated to an arbitrary filtering indeterminacy since filtered versions of original sources are still mutually independent. In practice, the ambiguity is not as dire as it seems and will depend on the statistics of the source. For temporally correlated data, in particular speech, the term “arbitrary filtering” ought to be replaced with “whitening”. We will explicate on this point in far greater detail later (Section 2.5, p. 36); for now it suffices to note that most of the distortion imparted by many signal separation algorithms comes in the form of temporal decorrelation.

Blind deconvolution algorithms can be divided into two main categories: those which perform all operations in the frequency domain, including the separation, and those which process the data in the time domain, using the frequency domain only for certain aspects (such as performing fast convolution). For Infomax species, a simple classification comes from examining the form of the nonlinearity $\varphi(\mathbf{u}) = \frac{g''(\mathbf{u})}{g'(\mathbf{u})}$. If $\mathbf{u}$ is a frequency domain quantity, then the separation optimization is executed in the complex frequency domain, relegating

it among the first category of algorithms. If **u** is a time-domain output, then separation is performed in the time-domain and the procedure belongs to the second class.

The first class of algorithms is rooted in the following idea:

Writing the convolutional mixing in terms of FIR polynomial matrices: [20]

$$\mathbf{X}(z) = \mathbf{A}(z) * \mathbf{S}(z)$$

A further decomposition results in:

$$\mathbf{X}_f(t) = \mathbf{A}_f(t) * \mathbf{S}_f(t).$$

This is an instantaneous signal separation problem for each frequency bin *f* for every short-time Fourier transform of the signal about the time point *t* [21]. Thus the convolutional problem can be broken up into many simpler problems, solvable by any of the methods outlined in Section 2.1. There is, however, an associated difficulty: permutation indeterminacy now appears at each frequency bin. A rigorous solution is still awaited, though there exist heuristic rules [21] and a few application-dependent methods [10] to overcome the problem.

Another difficulty, of pertinence only to Infomax algorithms, is what form the nonlinearity $\varphi(\mathbf{u}) = \frac{g''(\mathbf{u})}{g'(\mathbf{u})}$ takes in the complex domain. The question has analogues with the question of complex activation functions in neural networks [11], and a set of desiderata has been formulated in [22].

Due to such intractables, we do not investigate here frequency-domain algorithms, but devote space to two types of time-domain architectures.

2.3.1 Feedback Architecture

Torkkola in [23] addressed the problem of multichannel blind deconvolution with the following feedback network (shown in the 2x2 case), suitably generalised from Figure 4:

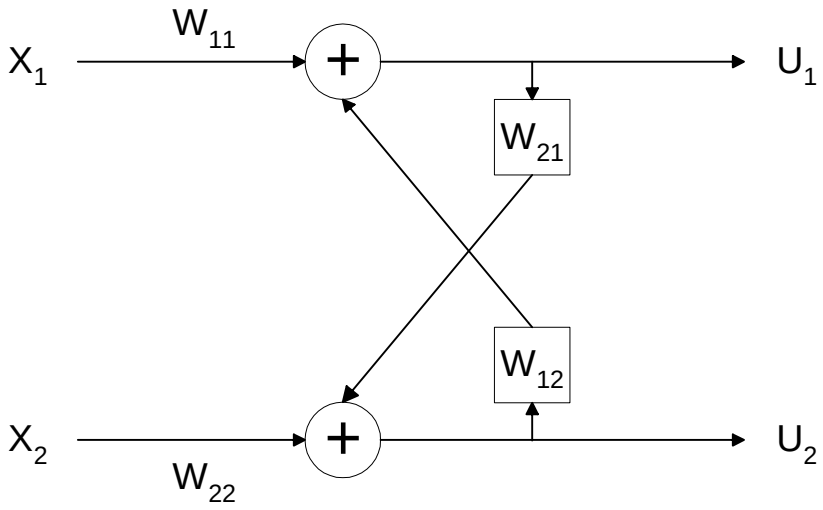


Figure 5: A feedback separation network employing adaptive cross filters

The output of this network is:

$$u_i(t) = w_i x_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{k=1}^{M} w_{ijk} u_j(t-k) \qquad \text{(Eq. 11)}$$

Here, only cross-filters w_{ij}, $i \neq j$, are full FIR filters performing separation; the direct filters w_{ii} are used, as in the case of the adaptive delay feedback system, for gain control [1]. This reduction of the direct filters to mere scaling coefficients ensures no filtering is applied to the reverberated version of each source, implying a reduction in the whitening effect (as well as any dereverberation). Whitening is also avoided in this configuration since the output of a cross filter is summed to a branch *different* to that of its input origin. Thus the algorithm removes redundancies across sources, and not within the source [23].

The Infomax algorithm relevant to a feedback deconvolution architecture may be computed by evaluating the gradient for, as usual, $J = \log\{p(g(\mathbf{u}))\}$. Consulting [23], we have as the final step:

$$\Delta w_{ii} \propto \frac{g''(u_i)}{g'(u_i)} x_i + \frac{1}{w_{ii}} \qquad \text{(Eq. 12)}$$

$$\Delta w_{ijk} \propto \frac{g''(u_i)}{g'(u_i)} u_j(t-k) \qquad \text{(Eq. 13)}$$

Lee [1] has noted the feedback architecture is only capable of separating minimum-phase systems: causal and stable mixings with causal and stable inverses. A room channel in which the echo is louder than the source signal gives rise to non-minimum phase transfer functions since there will exist zeroes lying outside the unit circle.

Another problem with feedback systems is the difficulty in incorporating instantaneous cross-filter feedback weights. Strictly speaking, the output **u**(t) is defined in terms of its present value as well as past values; however, this is nearly impossible to implement, requiring a host of computationally intensive special cases. Eq. 11 exhibits this difficulty, with cross filters beginning at time lag 1 instead of 0. Thus each "leading" cross weight is applied only about the mixtures at the previous time step, rendering it impossible to precisely cancel out interference at the current time point. Such a mechanism is not particularly damaging if the signal is temporally correlated (i.e. signal variation is small between adjacent time points), but fails completely in the case of white sources [1].

A feedforward system can also learn more general inverses than the feedback case, since feedforward FIR filters can approximate an inverse for a non-minimum phase mixing function. White sources are also not a problem here due to the lack of recursive definitions. We consider such an architecture next.

2.3.2 Feedforward Architecture

A feedforward FIR unmixing system produces outputs of the form:

$$\mathbf{u}(t) = \sum_{k=0}^{L-1} \mathbf{W}_k \mathbf{x}(t-k)$$

This corresponds with a matrix of FIR filters, displayed explicitly in the 2x2 case:

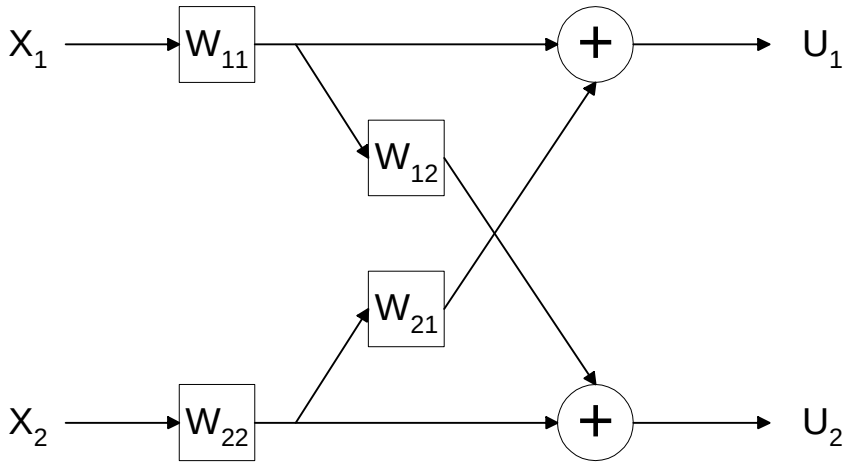


Figure 6: A feedforward matrix of filters

Here $\mathbf{W}_k$ denotes the usual matrix of elements W_{ijk} : *k*-th lag of the filter connecting the *j*-th source to the *i*-th microphone. Bell & Sejnowski in [16] derived a standard gradient algorithm for multichannel blind deconvolution by straightforward gradient calculation of the information-theoretic cost function.

We are interested mainly in the natural gradient algorithm, however, due to Amari *et al* [24]. Its derivation is based on examining the geometry of LTI filters in the *z*-domain: $\mathbf{W}(z) = \sum_{k=-\infty}^{\infty} \mathbf{W}_k z^{-k}$; because the natural gradient depends upon the parameter space, our information theoretic cost-function must also be formulated in the *z*-domain. The standard gradient $\nabla J(\mathbf{W}(z))$ which follows is then post-multiplied, not by $W^T W$ as in Section 2.1.3, but by $\mathbf{W}^T(z^{-1})\mathbf{W}(z)$ to form the natural gradient. A proof is omitted here, brave readers are referred to [24]; see also discussion in Section 2.4.1.

For the final stand, a computationally tractable update is given in [24]:

$$\Delta \mathbf{W}_k = \mathbf{W}_k + \varphi(\mathbf{u}(t-L+1))\mathbf{p}^T(t-k) \qquad \text{(Eq. 14)}$$

where $\mathbf{p}(t) = \sum_{q=0}^{L-1} \left(\mathbf{W}_{L-1-q}\right)^T \mathbf{u}(t-q)$ (Eq. 15)

These two equations constitute the Natural Gradient Infomax Feedforward algorithm for convolved sources.

2.3.3 Block Implementations

The natural gradient algorithm as delineated above updates the un-mixing filters every time a new sample vector **x** is presented; an alternative strategy is to average the data over a block $[\mathbf{x}(t),...,\mathbf{x}(t+M)]$ and then update **W**. Though this slows the rate of convergence in real-time, there exist at least two reasons for such implementation:

1) FIR block adaptive filtering can be employed efficiently in the frequency domain.

2) The use of blocks to average data increases the stability of an algorithm by minimizing the influence of random variations in the signal. [1]

Na *et al* [25] outline the details for producing a block implementation of Amari's natural gradient algorithm. This involves writing the update as an averaged sum of data, then using well-known overlap-save methods as outlined, for instance in [11], to compute fast correlations and fast convolutions. It is important to note that this is still a *time*-domain algorithm, since φ is formulated in the time-domain; the frequency-domain is only brought to bear as a useful aid in filtering.

We present a succinct summary of the Block Natural Gradient Algorithm; technical details may be found in [25].

1. Initialize the un-mixing filters in the frequency domain, $\mathbf{W}(b)\big|_{b=0}$, where *b* stands for block number.

2. Given a deconvolving filter length of *L*, compute the Fourier transform (all transforms are length 2*L* for 50% overlap) of two blocks of the input:

$$\mathbf{X}(b) = \mathbf{F}\{[\underbrace{\mathbf{x}((b-1)L,....}_{(b-1)\text{-th block}},\underbrace{\mathbf{x}((b+1)L-1)}_{b\text{-th block}}]\}$$

3. Calculate the frequency-domain output $\mathbf{U}(b)$:

$U_i(b) = \sum_{j=1}^{n} \mathbf{W}_{ij}(b) \circ \mathbf{X}_j(b)$, where $\circ$ denotes the component-wise multiplication between two vectors.

4. Compute the b-th block of the time-domain output $\mathbf{u}(bL)$:

$u_i(bL) = \text{last } L \text{ elements of } \mathbf{F}^{-1}\{U_i(b)\}$

5. Compute $\mathbf{p}(bL)$:

$p_j(bL) = \text{first } L \text{ elements of } \mathbf{F}^{-1}\{\sum_{i=1}^{n} (\mathbf{W}_{ij}(b))^* \circ \mathbf{Y}_i(b)\}$, where * denotes complex conjugate.

6. Form the frequency-domain version of $\mathbf{p}$:

$$\mathbf{P}(b) = \mathbf{F}\{[\underbrace{\mathbf{p}((b-1)L, \ldots}_{(b-1)-\text{th block}}, \underbrace{\mathbf{p}((b+1)L-1)}_{b-\text{th block}}]\}$$

7. Compute the temporary frequency domain quantity:

$$\mathbf{\Phi}(b) = \mathbf{F}\{[\underbrace{0, \ldots, 0}_{L \text{ zeros}}, \varphi(\mathbf{u}(bL)), \ldots, \varphi(\mathbf{u}((b+1)L-1))]\}$$

8. Calculate the gradient:

$\Delta\mathbf{W}_{ij}(bL) = \text{first } L \text{ elements of } \mathbf{F}^{-1}\{\mathbf{W}_{ij}(b) + \mathbf{\Phi}_i(b-1) \circ \mathbf{P}_j^*(b)\}$

9. Update the filters:

$$\mathbf{W}_{ij}(b+1) = \mathbf{W}_{ij}(b) + \mu\mathbf{F}\{[\Delta\mathbf{W}_{ij}^T(bL), \underbrace{0, \ldots, 0}_{L \text{ zeros}}]\}$$

10. Return to step 2, incrementing the block number.

This concludes our overview of significant algorithms for Blind Signal Separation. Readers acquainted with the field will note that we have not said anything about the plethora of deconvolution procedures using only second-order statistics. Lack of space and time has not permitted the inclusion; they are, however, a straightforward generalization of the concepts presented for the instantaneous case in Section 2.1.4.

2.4 Optimization Strategies

Speed of convergence is a highly important issue for online adaptation, even in slowly changing environments. Since Independent Component Analysis reduces to an optimization problem for the vast majority of cases, it is useful to examine possible approaches for solving the system of equations: $\frac{\partial J}{\partial \mathbf{W}} = \mathbf{0}$. Except for Section 2.4.3, we do not specifically refer to the Infomax technique but rather any algorithm based on cost function optimization.

2.4.1 The Natural Gradient

Consider the Euclidean vector space $\mathbb{R}^n$. Define a cost function $J : \mathbb{R}^n \to \mathbb{R}$. It is well-known that the gradient $\nabla J(\mathbf{x}) \equiv \left(\frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2}, ..., \frac{\partial J}{\partial x_n} \right)$ gives the direction of steepest-ascent at the point **x,** making it a prime candidate for the search direction[8]. However, what is not well-known is that if one can formulate the cost function in terms of *nxn* matrices (i.e. for even n, $J : \mathbb{R}^{\frac{n}{2} x \frac{n}{2}} \to \mathbb{R}$), the direction of steepest-ascent is *not* given by ∇J, but by the *natural* gradient, a quantity due to Amari.

A simple example concretizes the distinction:

Let $f(w_1, w_2, w_3, w_4) = \ln | w_1 w_4 - w_2 w_3 |$. One can easily compute the gradient as:

$$\nabla f = \left(\frac{w_4}{w_1 w_4 - w_2 w_3}, \frac{-w_3}{w_1 w_4 - w_2 w_3}, \frac{-w_2}{w_1 w_4 - w_2 w_3}, \frac{w_1}{w_1 w_4 - w_2 w_3} \right) \qquad \text{(Eq. 16)}$$

However, we may also formulate *f* as a matrix function:

---

[8] [26] has noted that the conjugate gradient search direction, which also only makes use of gradient information, is often more efficient.

$f(\mathbf{W}) = \ln|\det(\mathbf{W})|$, whose gradient is:

$$\nabla_{\mathbf{W}} f = (\mathbf{W}^T)^{-1}, \qquad \text{(Eq. 17)}$$

a consideration in complete agreement with Eq. 16 under the mapping

$$w_{11} = w_1,\ w_{12} = w_2,\ w_{21} = w_3,\ w_{22} = w_4.$$

Equation 17 is not the steepest ascent direction in the matrix space, however. It may appear we have simply changed notation, but we have really changed spaces: from that of vectors to that of matrices. The latter possess a *group* structure which the former lacks: a multiplication structure. This (Lie) group structure modifies the very geometry of the parameter space; the set of variables behave like a curved manifold with a metric tensor $\mathbf{g}(\mathbf{W})$.

Of course in this new space the standard gradient cannot provide the direction of steepest ascent, valid as it was only in a Euclidean geometry. Amari in [17] proved by Lagrange multipliers that the direction of steepest ascent is given by:

$$\nabla^{\mathbf{W}} f = \mathbf{g}^{-1}(\mathbf{W}) \cdot \nabla_{\mathbf{W}} f$$

This is the natural gradient. For those familiar with the terminology of differential geometry, $\nabla^{\mathbf{W}} f$ is the *vector* gradient or contravariant gradient, whereas $\nabla_{\mathbf{W}} f$ is the *one-form*, or covariant gradient. The two are duals of one another.

Amari analyzed the form the metric tensor assumes in a variety of contexts, such as parameter estimation of probability distributions and multilayer perceptrons in a neural network, finding that the geometric space often attains a Riemannian character. In both cases the Fisher Information Matrix produces a metric. For instantaneous Blind Signal Separation, where the parameter space is the set of non-singular *nxn* matrices, he proved in [17] that $\nabla^{\mathbf{W}} f = \nabla_{\mathbf{W}} f \cdot \mathbf{W}^T\mathbf{W}$, whereas for convolutive separation the required modification is: $\nabla^{\mathbf{W}} f = \nabla_{\mathbf{W}} f \cdot \mathbf{W}^T(z^{-1}) \cdot \mathbf{W}(z)$ [24].

The moral behind this discussion is that if one can equip the parameter space with a richer structure (such as the matrix algebraic structure), convergence speed can be increased. Empirical experiments have shown that the natural gradient is vastly superior to the standard gradient for optimizing matrix cost functions [27].

2.4.2 Newton Iteration

Another interesting path is afforded by Newton's Method. Suppose we seek the solution of:

$$\mathbf{F}(\mathbf{W}) = \mathbf{0}, \text{ where } \mathbf{F} = \frac{\partial J}{\partial \mathbf{W}}.$$

Newton's method states the answer can be found recursively via:

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \frac{\mathbf{F}(\mathbf{W}(n))}{\mathbf{dF}(\mathbf{W}(n))}$$

This idea was implemented by Hyvarinen in his misnomered "A fast fixed-point algorithm for ICA", (it is a Newton algorithm) [28]. Hyvarinen's **F** consisted of the usual algebraic system obtained under Lagrange extremum conditions to a cost function "approximating negentropy". The advantage of using Newton's Method is that the procedure usually gives at least quadratic convergence. However, because there is no step-size parameter in the update, the algorithm can be unstable, completely eliminating the possibility of a non-averaged (non-batch) based adaptation. Thus the "fixed-point algorithm" is practical only in off-line applications.

2.4.3 Choosing the Nonlinear Activation Function "g"

It is a peculiarity of Infomax algorithms that successful separation depends upon a suitable choice of the function $g(u)$. Recall from Section 2.1.2 that we must have $g'(\mathbf{u}) = p(\mathbf{u})$, where the right hand side denotes the (global) probability distribution function of the separated outputs.

A surprising fact, backed by numerous simulation results performed by researchers, is that ICA algorithms with a fixed nonlinearity converge to a separating solution despite the fact that the nonlinearity is often a crude approximation to the underlying source distribution [1]. In other words, the choice of *g* is somewhat flexible.

A theoretical argument for such robustness, first hinted by Cardoso [29], suggests model-mismatch is tolerable because sources are recovered only up to scaling factors. Consider for instance the family of sigmoidal nonlinearities: $g(u) = \lambda \tanh(\alpha u)$, where λ and α are tuneable parameters. This family is the most commonly employed of all "squashing" functions. Its derivative is displayed in the following figure:

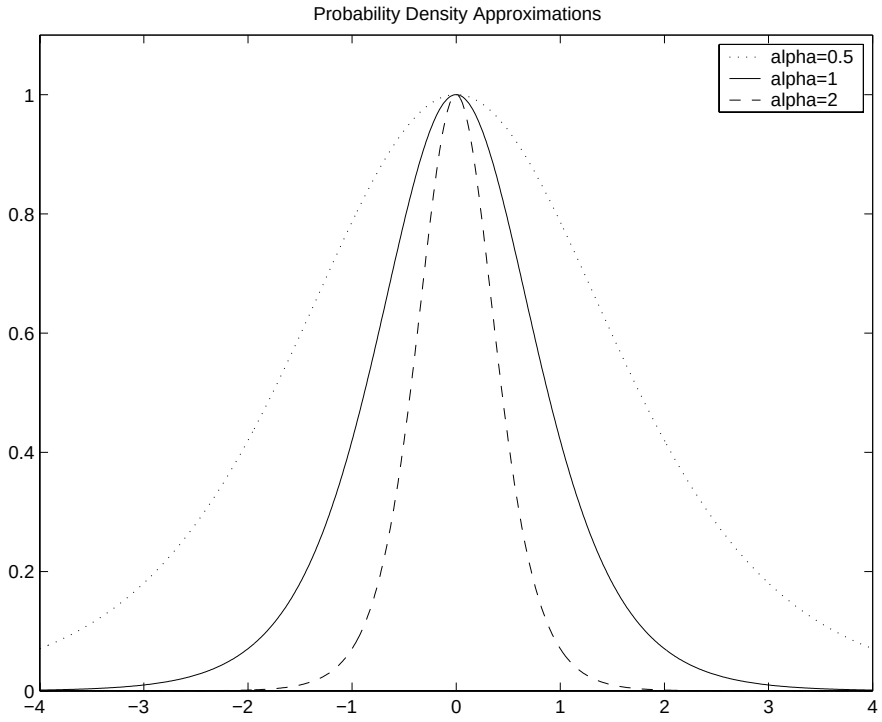


Figure 7: Graph of $\text{sech}^2(\alpha u)$ with varying α

As can be plainly seen, the significance of α resides in controlling the variance of the distribution. Together with the scale factor λ, (which normalises the function to a proper probability density function), typical signal histograms such as speech or music may be modeled somewhat accurately by one or two parameters. However, there is no need to model α, since the variance can be adjusted arbitrarily via amplitude scaling of the sources – one of the

ambiguities in source recovery. Moreover, the form of the update in Infomax algorithms does not use g, but rather $\varphi(\mathbf{u}) = \frac{g''(\mathbf{u})}{g'(\mathbf{u})}$. With $g(u) = \lambda \tanh(\alpha u)$, we have $\varphi(u) = -2\alpha \tanh(\alpha u)$. λ vanishes in the final update: there is no need to model the scaling factor λ either. This two-parameter freedom is precisely the flexibility required in model-mismatch robustness.

Still, one cannot afford excessive laxity in the choice of nonlinearity. An ill-matched squashing function may converge to the correct solution, but will require many more iterations than a well-matched density.

A further consideration comes from the choice of a density family. For the separation of speech signals at least, the Laplacian distribution $e^{-\alpha|u|}$ is more appropriate than $\operatorname{sech}^2 u$, obtaining the proper convexity profile and center cusp [30]:

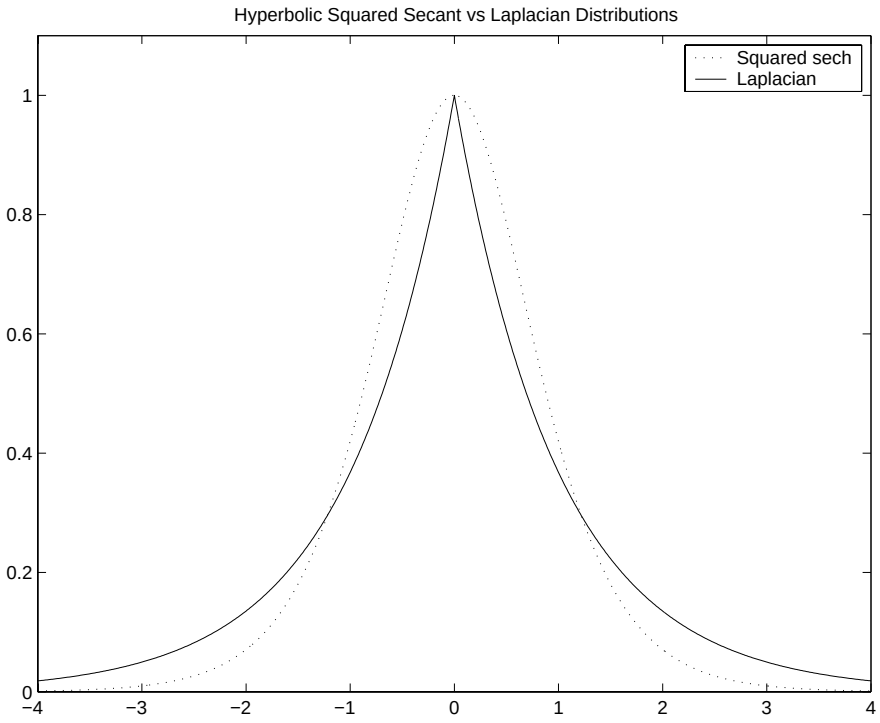


Figure 8: Comparison of Laplacian and $\operatorname{sech}^2 u$ distributions

An additional pleasantry occurs with the Laplacian distribution: the form of the score function $\varphi(u)$ inhabits a beautifully simple form:

$$\varphi(u) = -\alpha \,\mathrm{sgn}(u)$$

The parameter α may be obtained by estimating the source variances $Var_{sources}$ and then matching with the Laplacian variance:

$$Var_{sources} = Var[X_{Laplacian}] = \frac{2}{\alpha^2}$$
$$\rightarrow \ \alpha = \sqrt{\frac{2}{Var_{sources}}} \qquad \text{(Eqs. 18)}$$

Even without knowledge of source variances one can obtain an estimate by globally computing the received mixture variances.

Interesting physical insight may be gleaned by examining the direct form of *g*:

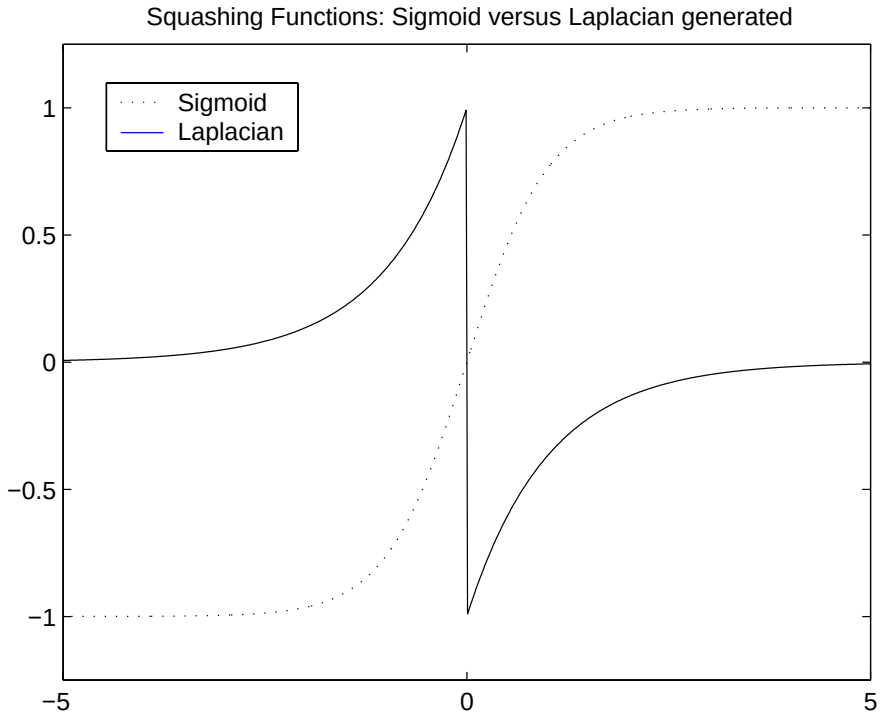


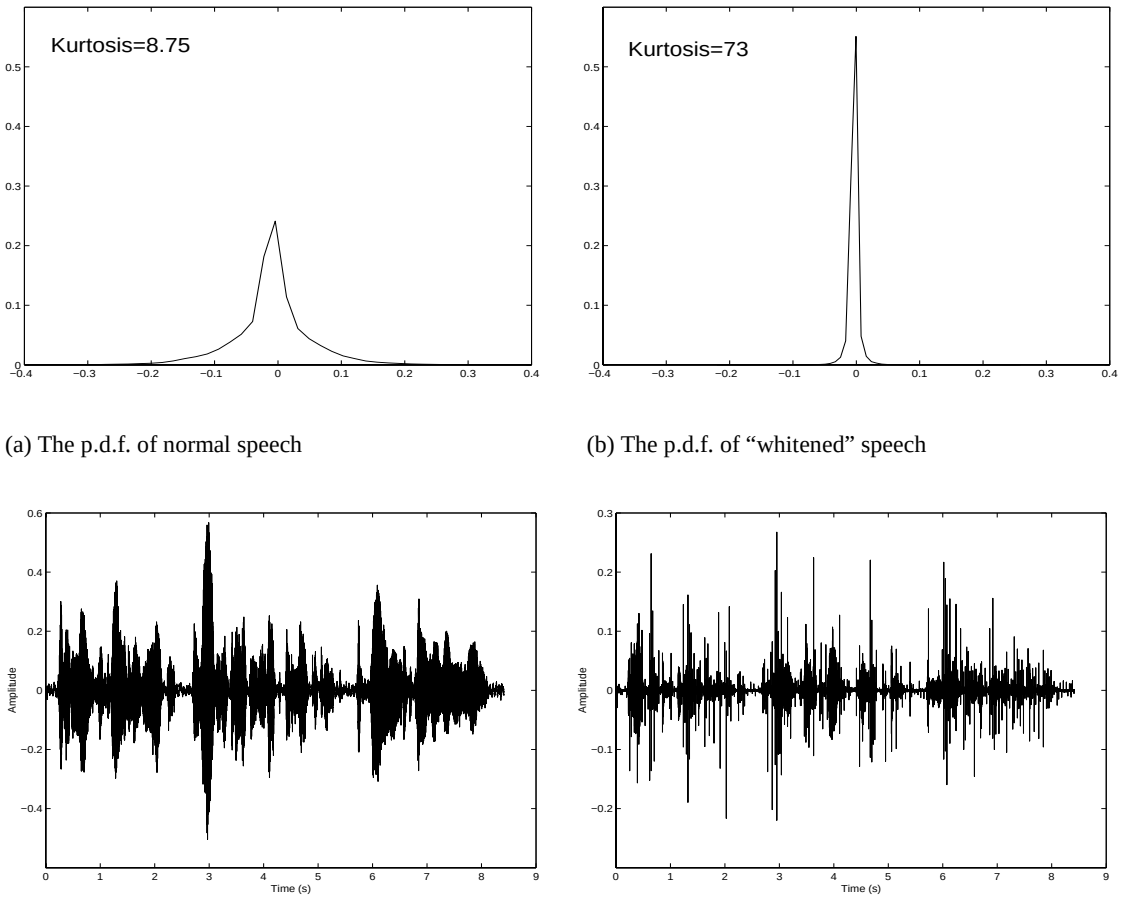
Figure 9: Comparison of nonlinearities: $g(u) = \int \mathrm{sech}^2 u \; du$ and $g(u) = \int e^{-|u|} \; du$

The above graph demonstrates the role of the activation function as an invertible amplitude bounder – necessary so that entropy maximization of $g(u)$ does not diverge.

2.5 Whitening and a LP Residual-Domain Weight Update

It has been well-documented that for self-correlated inputs (speech being a prime example), practically all time-domain blind deconvolution algorithms exhibit the side-effect of *whitening* [1, 23]. This corresponds with a flattening of the power spectrum: energy at higher frequencies is increased at the expense of energy in lower frequency bands.

Why is this so? After all, blind signal separation is supposed to perform spatial decorrelation, not temporal decorrelation. The answer resides in the following figures:



(a) The p.d.f. of normal speech

(b) The p.d.f. of "whitened" speech

(c) Normal speech – time domain

(d) 10-th order LP residual of speech

Fig 10: Comparison of probability density functions and kurtosis for normal and whitened speech

Fig.10 displays a histogram and time-evolution of normal speech on the left side, and a histogram and time-evolution of "whitened speech" on the right side. "Whitened speech" may be simulated by computation of an *n*th order Linear Prediction (LP) residual, which represents a speech signal with the short-term correlation removed. The diagrams above show that whitened speech has far smaller amplitude spread than normal speech, resulting in a drastically higher sample kurtosis. Since kurtosis gives an approximate indication of negentropy (distance from the Gaussian), which from previous discussion is equivalent to information maximization, log-likelihood maximization etc., we see that adaptive filtering employing typical measures of independence must necessarily time-decorrelate as well as spatially decorrelate.

Torkkola in [23] already noted the deleterious effects of whitening. In his experiments the joint entropy increase due to whitening was so large as to overshadow entropy increases due to spatial separation. Thus weighting coefficients converged to mere whitening filters, and not separating filters. In our own experiments (Section 3.3) the whitening effect was significant enough to destroy convergence. Torkkola's solution was to introduce a feedback structure where the direct filters were reduced to scaling coefficients (Section 2.3.1). However, feedback structures have special weaknesses not present in feedforward architectures, as already mentioned (Section 2.3).

We present here a novel method to eliminate whitening in the speech separation process, while preserving feedforward design. The key idea involves a type of temporal *pre-whitening* of the mixed speech signals via LP analysis filters. ICA is then performed in the residual domain, and a synthesis filter reconstructs the separated speech from the residuals. We expect the following theoretical improvements to result:

- Since LP residuals have most of the short-term correlation removed, there can be little to no further entropy increase due to temporal decorrelation: ICA will follow those directions which separate rather than whiten.

- The short-term uncorrelatedness of the LP waveform will result in a more stable algorithm since adjacent weight updates become independent of one another.

The second point may be realised by thinking of stochastic search directions as deviations from some preferred deterministic locus. *Correlated* weight updates tend to reinforce these deviations, resulting in oftimes quicker convergence but generally unstable (high and low amplitude) behaviour. By performing filtering on (temporally) pre-whitened samples, weight updates are rendered independent of one another. Our process may then be thought of as introducing an *anti-momentum* (stabilizing negative feedback) term to the convergence.

Use of LP analysis for adaptive filtering may seem unusual given its traditional role in speech coding. There are precedents, however, in the domain of speech enhancement [31]. Gillespie *et al* [32] have applied a kurtosis-based cost function to the LP residual for dereverberation.

The following displays a theoretical setup:

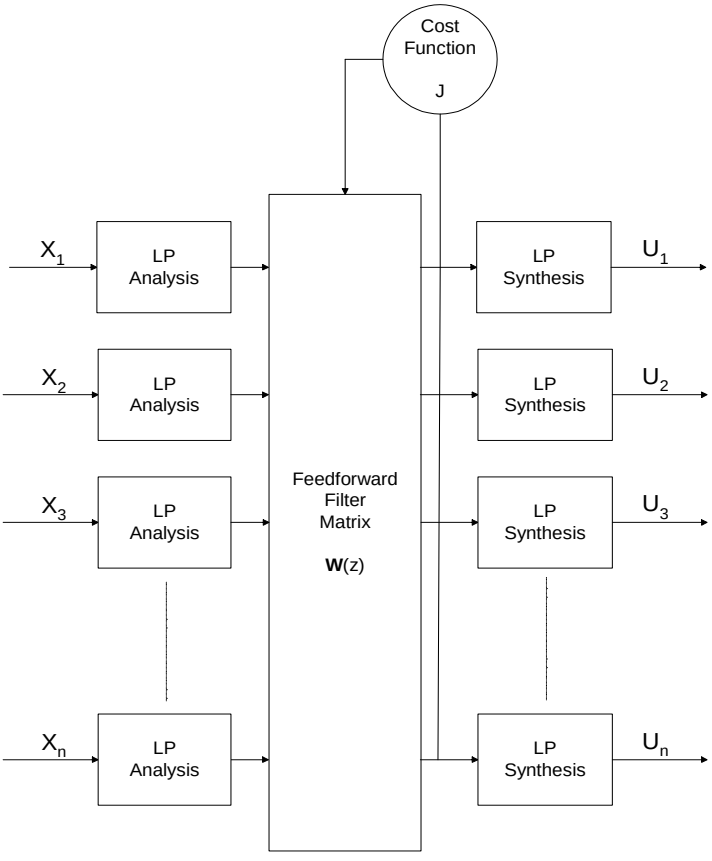


Fig 11: ICA architecture on speech residuals

There exists a problem with the above configuration, however: the inverse LP filter introduces reconstruction artefacts. It is important to realise that these artefacts arise not because of block-edge effects, but rather to the fact that the output speech is necessarily different from the input speech, due to adaptive filtering. LP coefficients from initial analysis *cannot* be used to recover speech since they are appropriate coefficients only for the signal before the filter. Malvar *et al* [32] has noted this effect in the case of dereverberation; for blind signal separation the distortion is bound to be even more pronounced since LP analysis is performed on not one, but a sum of *n* speakers, while the output is supposed to represent but one speaker. A way to circumvent the problem is to adapt coefficients in the residual domain, but actually apply the filter to the time-domain signal *without modification.* This is essentially the solution contained in [32], which was justified by the assumption of a linear system with small step-sizes. Our justification for ICA lies in the hope that LP residuals retain enough significant speech information so the extremum of the cost function remains invariant under analysis filtering. This can be tuned by modifying the LP filter order. With this variation, Fig. 11 attains the form:

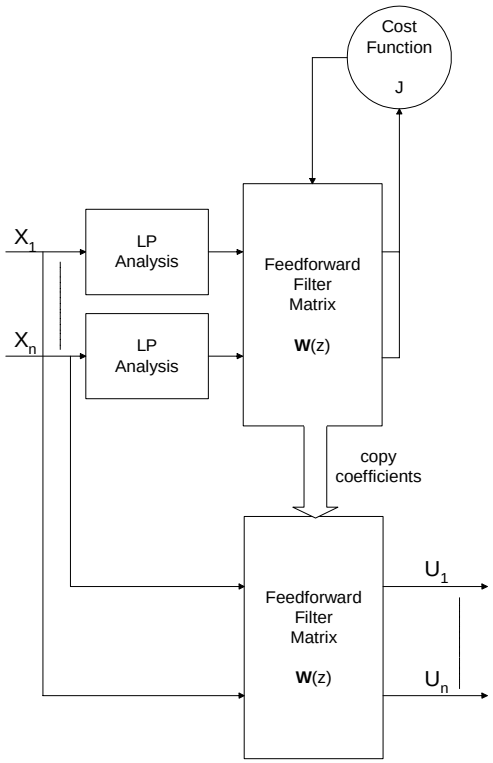


Fig 12: Equivalent ICA architecture without residual reconstruction artefacts

We refer to the above as ICA with a LP residual-domain weight update. Experimental results for a time-domain version of this system are given in Section 3.4. Though we illustrate its utility only with the Infomax cost function, the technique may be applied to any blind signal separation algorithm utilising feedforward adaptive filtering.

3 Experimental Results

The corpus of results which follow represent the culmination of numerous simulations performed in Matlab. For brevity we have not displayed *all* tests but rather selected those deemed germane. Experiments were conducted with speech segments sampled at 16 kHz, lasting approximately 9 seconds, obtained from the Lincoln Laboratory Speech Enhancement Corpus (LLSEC). No signal pre-processing was used prior to applying a separation algorithm. Simulations were performed on a 1 GHz Pentium III machine.

Absolute necessary to navigate the maze of results is the audio guide included in Appendix B, which provides a map between the disk directory accompanying this report and the sections contained herein.

3.1 Instantaneous Mixtures

Here, nxn channel matrices were generated to artificially mix the sources.

Since the mixing matrices $\mathbf{A}$ are known beforehand, it is easy to determine separation success by examining a "performance" matrix $\mathbf{P} = \mathbf{WA}$. Perfect separation renders $\mathbf{P}$ a scaled and permuted version of the identity. The following cross-talk error measure is invariant under such scaling and permutation:

$$\varepsilon = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{|p_{ij}|}{\max_k |p_{ik}|}-1\right)+\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{|p_{ij}|}{max_k |p_{kj}|}-1\right)$$

Note for perfect separation, (up to standard indeterminacies) $\varepsilon = 0$.

3.1.1 Natural Gradient Algorithm

The algorithm of Eq. 1 was used to separate 4 speech sources mixed under 20 different randomly generated (uniform on $[0,1]$) matrices. Parameters were fixed as follows: learning rate $\mu = 0.0005$, nonlinearity $\varphi(\mathbf{u}) = -2\tanh(\mathbf{u})$, and $\mathbf{W}(0) = \mathbf{I}$. An update was performed

for every incoming speech sample, thus for 9 seconds of speech or 1.35 x 10^5 samples. Displayed below are the convergence loci:

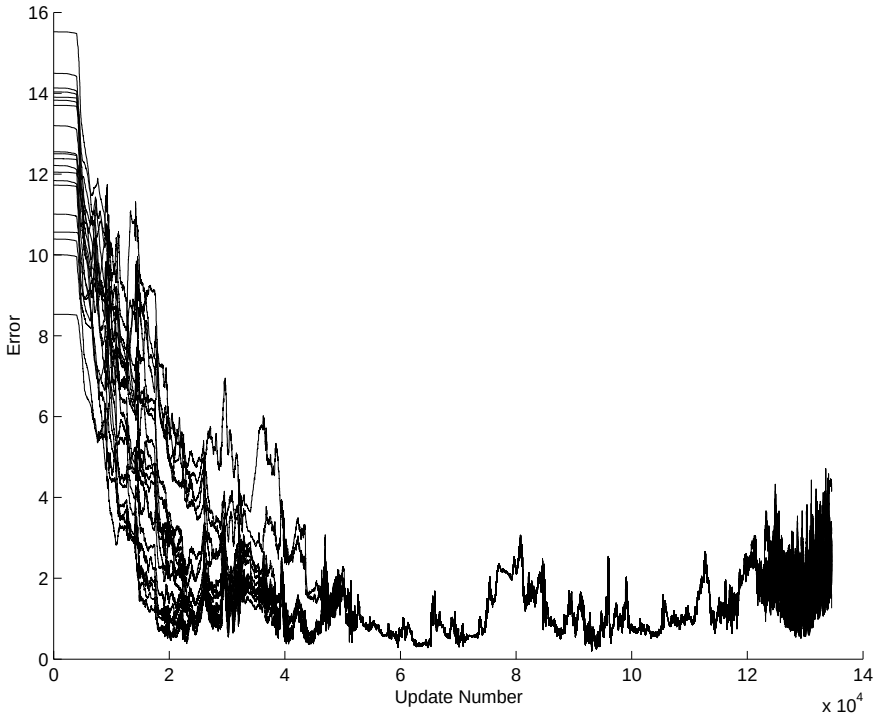


Figure 13: Convergence curves for the Instantaneous Natural Gradient Algorithm under Random Channel Conditions

As one can see, despite the variety of initial mixings the algorithm converges in a relatively uniform manner (the so-called equivariance property [27]). Separation of a single mixing scenario required only 10.845 seconds of computation time for 9 seconds of speech – very nearly real-time separation. Informal listening tests reveal an excellent quality of separation.

3.1.2 Optimal Nonlinearities

The above simulation utilitized the squashing function $g(u) = \lambda \tanh(\alpha u)$, with $\lambda = \alpha = 1$, representing an unmatched nonlinearity (see Section 2.4.3). In this section we compare its performance with the more theoretically optimal squashing function $g(u) = \int e^{-\alpha|u|}\, du$, designed to accurately reflect the Laplacian character of speech. The score function in turn becomes $\varphi(u) = -\alpha \operatorname{sgn}(u)$, where α is automatically chosen via Eq. 18 and

by evaluating mixture variances. The following graph shows two convergence curves of the natural gradient algorithm under identical mixing scenarios, one under the standard sigmoidal squashing function and the other with the matched Laplacian squashing function.

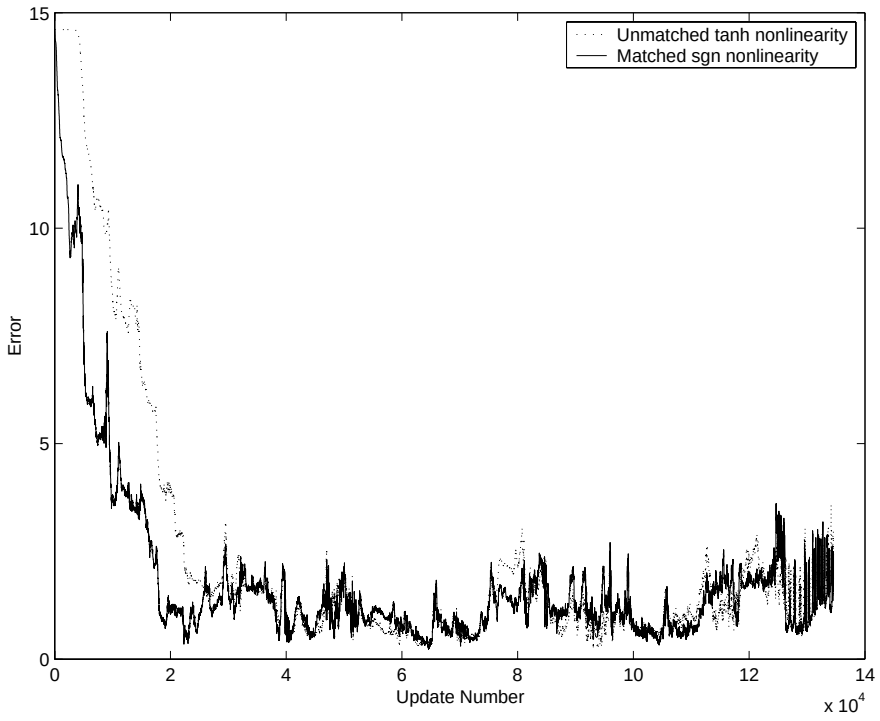


Figure 14: Comparison of convergence curves with unmatched sigmoidal and matched Laplacian nonlinearites

Clearly the matched nonlinearity outperforms the unmatched nonlinearity, converging nearly twice as quickly.

We also give, in analogy with Figure 13, the convergence curves of the matched algorithm under 20 randomly generated mixing scenarios:

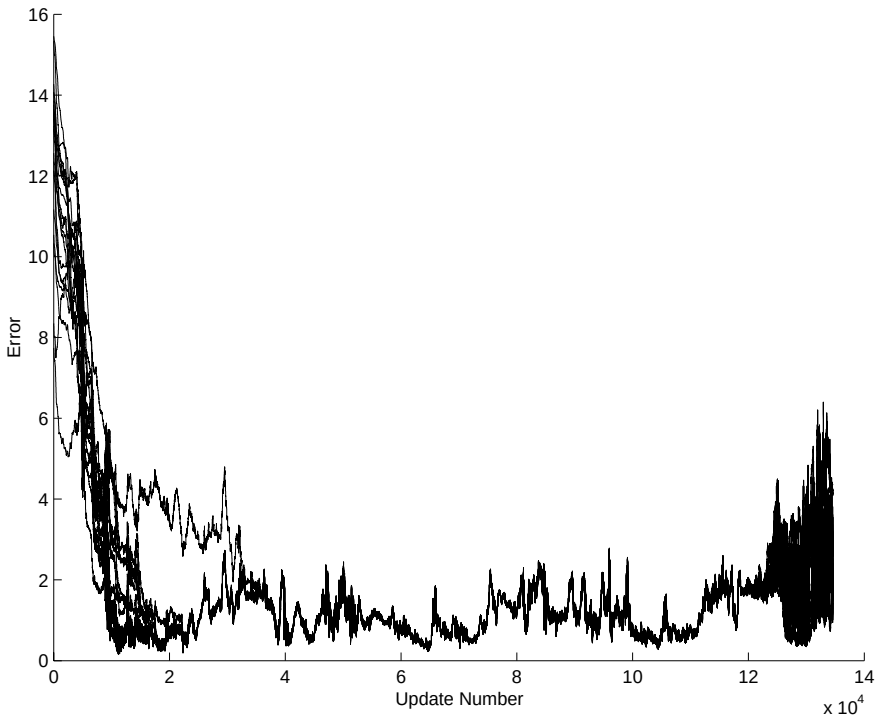


Fig 15: Convergence curves for a matched Natural Gradient Algorithm under Random Channel Conditions

The above is plainly superior to its counterpart (Fig. 13).

3.1.3 Effect of a time-varying mixing matrix

Here, we investigate the ability of the natural gradient algorithm to track moving speakers. Begin with the following initial mixing matrix:

$$\mathbf{A} = \begin{bmatrix} 0.3046 & 0.3028 & 0.3784 & 0.4966 \\ 0.1897 & 0.5417 & 0.8600 & 0.8998 \\ 0.1934 & 0.1509 & 0.8537 & 0.8216 \\ 0.6822 & 0.6979 & 0.5936 & 0.6449 \end{bmatrix}$$

From this point, 0.000005 was subtracted off every element of the first column at every sampling moment, until the elements reached 0. Over the course of 9 seconds of speech, the mixing matrix attains a final form of:

$$\mathbf{A} = \begin{bmatrix} 0 & 0.3028 & 0.3784 & 0.4966 \\ 0 & 0.5417 & 0.8600 & 0.8998 \\ 0 & 0.1509 & 0.8537 & 0.8216 \\ 0.0323 & 0.6979 & 0.5936 & 0.6449 \end{bmatrix}$$

This alteration physically corresponds with the first speaker gradually moving away from all microphones until there are practically only 3 sources. An error curve is presented next, with parameters *ceterus paribus* to those of Section 3.1.1:

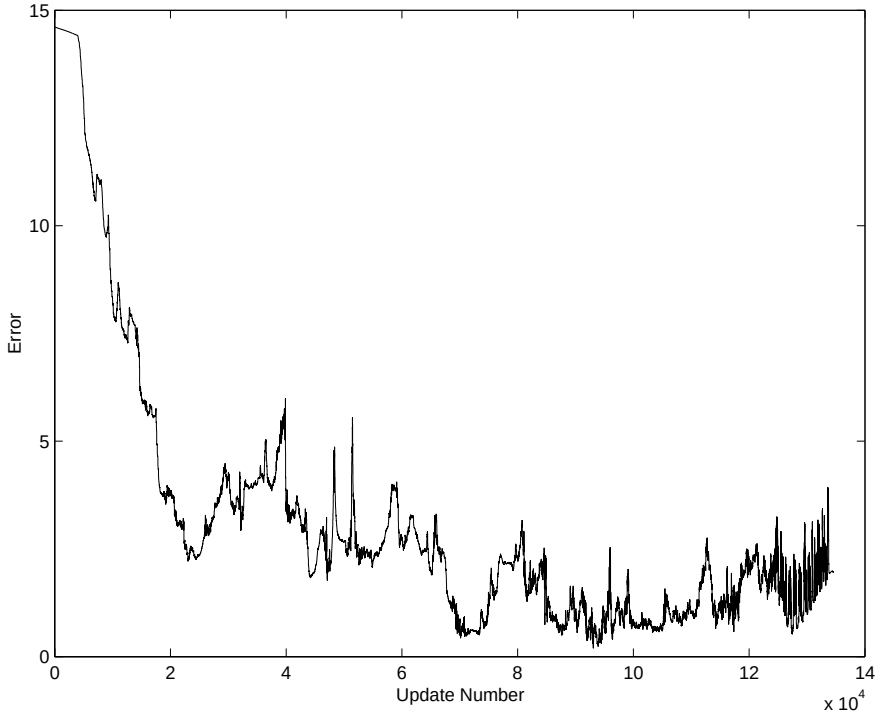


Figure 16: Convergence curve of natural gradient algorithm under fading source mixing

The reader is urged to examine for himself the quality of separation by observing the audio waveforms. A somewhat satisfactory separation seems to occur at 2 seconds on all channels, but at approximately 3.5 seconds, two channels "switch" with one another. This seems to be an interesting manifestation of the permutation ambiguity in separation.

3.1.4 Multiple decorrelation

We apply the multiple decorrelation algorithm of Equations 3 & 4 for the separation of instantaneously mixed voices. In its present incarnation, the procedure is not a local adaptive algorithm but depends upon a measurement of K cross-covariances to estimate the channel matrix **A**. The algorithm derived in Section 2.1.4 relied upon the presumed non-stationary qualities of sources. For speech this involves calculating the cross-covariances at distinct inter-frame points via the average of local intra-frame covariances. Two conflicting ideals are present: the sample size required to estimate each cross-covariance must be large for accurate estimation, however, it cannot be too large so as to overstep the interval in which speech possesses a static instantaneous covariance function (5-20 ms). Parameters were established as follows:

- $K = 13$ cross-covariance measurements $\mathbf{R}(k)$ in total, one at every 10000 sample interval, by averaging 400 local instantaneous cross-covariances.
- In accordance with the theoretical dictates of Section 2.1.4, diagonal elements of **A** set to 1, removing the scaling indeterminacy.

Because the algorithm is an *offline* deterministic procedure, it is not possible to directly compare the performance with that of the natural gradient. In particular, the reliance on deterministic optimization renders the convergence curve an inadequate description of performance, since the rate of convergence can be nearly arbitrarily set by increasing the step size μ:

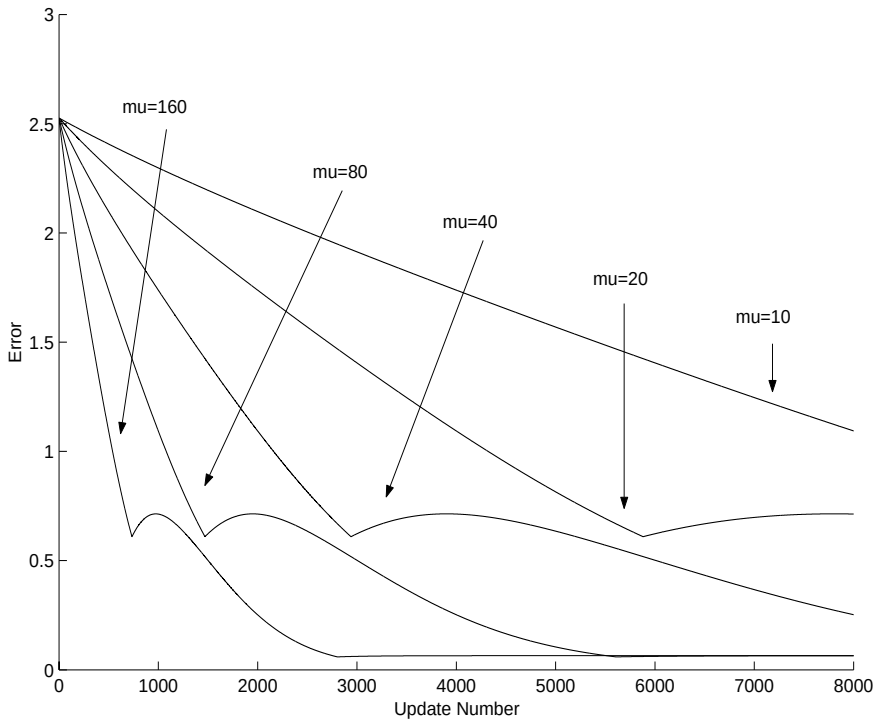


Fig 17: Error curve of multiple decorrelation algorithm on two mixed voices with changing gradient step-size

The 8000 iterations shown above required 24.025 seconds of processing time. The resulting waveforms sound perfectly and uniformly separated, since the inverse matrix $\mathbf{W} = \mathbf{A}^{-1}$ is only applied to the signals *after* optimization.

What follows are convergence curves for the separation of two voices under random mixing, using the fixed step size $\mu = 80$:

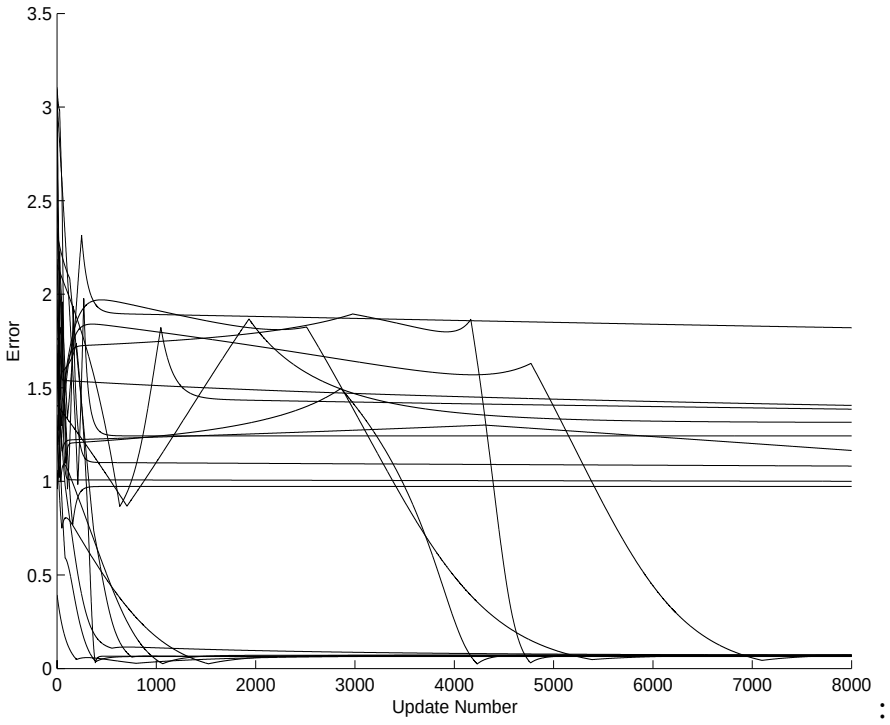


Fig 18: Error curves of multiple decorrelation algorithm for two mixed voices under random channel conditions

The result shocking: fully half of the mixings do is not converge. This is experimental confirmation that equivariance is not one of the properties of multiple decorelation. Another weakness rears its head in the separation of more than two voices: under the initial 4-voice matrix of Section 3.1.3, the algorithm converges to a spurious minimum:

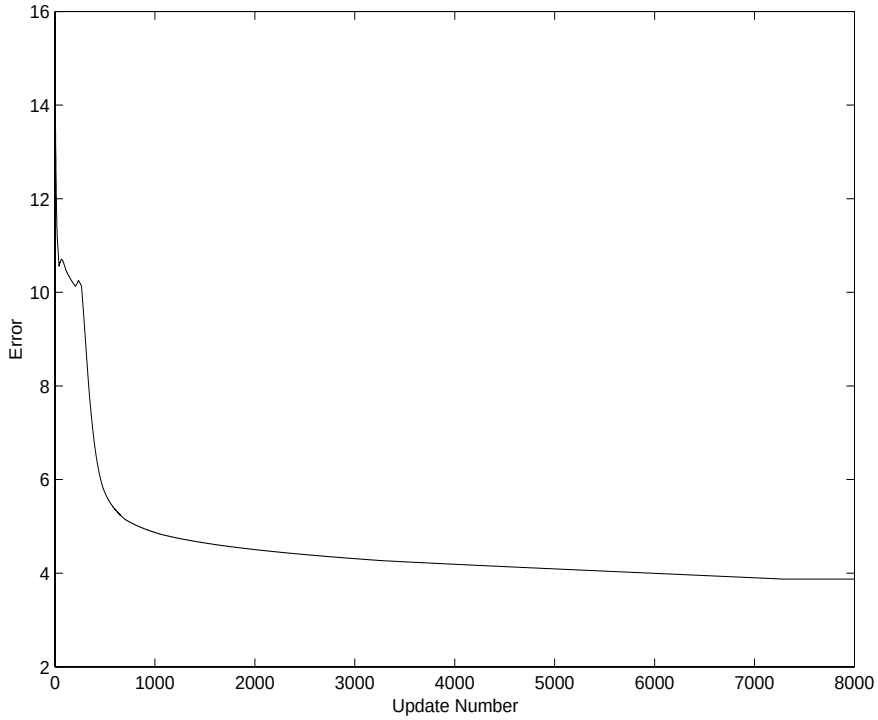


Figure 19: Error curve of multiple decorrelation algorithm for the separation of 4 voices

A quick listen to the audio file confirms completely inadequate separation.

We conclude that on nearly all counts the natural gradient algorithm is far superior to the multiple decorrelation algorithm. The Infomax procedure is local, adaptive, independent of the channel (equivariant), and handles a relatively large number of sources well, whereas the latter is deficient in all these respects.

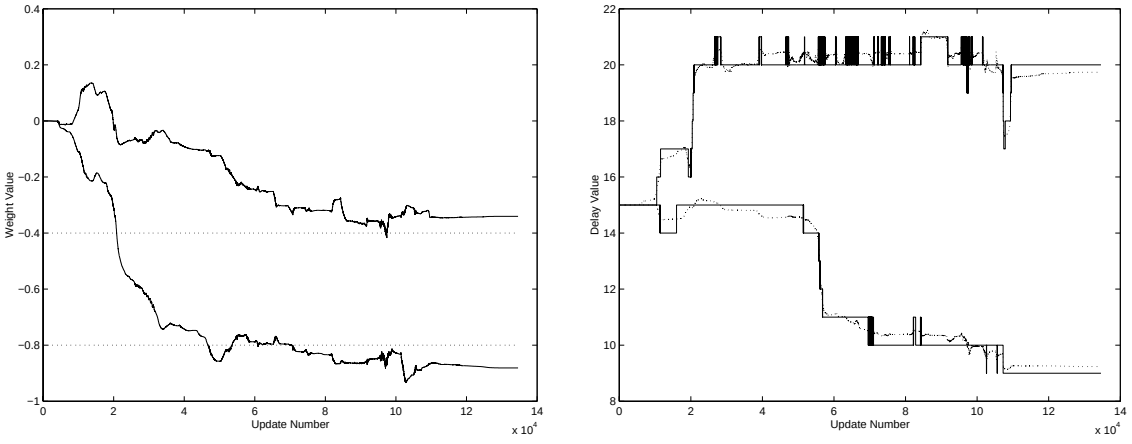
3.2 Delayed Mixtures

Torkkola's adaptive delay system of Fig. 4 and Eqs 7-9 are implemented. Two speech sources were mixed according to the formulas:

$$x_1(t) = s_1(t) + 0.4s_2(t-10)$$
$$x_2(t) = s_2(t) + 0.8s_1(t-20)$$

We do not implement the direct delay, but only the relative delay, in compliance with discussion in Section 2.2. An update was performed at every speech sample, for all 9 seconds of speech. This required 25.116 seconds in total processing time.

With initial settings $w_{11} = w_{22} = 1,\ w_{12} = w_{21} = 0,\ d_{12} = d_{21} = 15$, a w_{ij} learning rate of $\mu = 0.0005$ and a delay learning rate of $\nu = 0.1$ the following parameter convergences occurred:



(a) Cross-Weights (b) Cross-delays (dashed), rounded cross-delays (solid)

Fig 20: Convergence of parameters for feedback adaptive delay system

The second graph gives both an internal, floating point delay value of the update algorithm and the rounded delay value (delay must be an integer before application to mixtures).

From previous work, we know that cross-weights should converge to the negative of the mixing cross coefficients: 0.4 and 0.8. Adaptive delays should also learn actual mixing de-

lays: 10 and 20. Both conditions are satisfied but convergence is not very strong. A listening test confirms that there is separation, but only after approximately 6 seconds of speech.

The problem with an adaptive delay algorithm is that to achieve separation, delays must be estimated highly accurately. Without perfect estimation, the weights have a difficult time converging to appropriate values. A deviation of just 1 sample caused separation problems.

Of even greater concern is the very erratic and non-robustness of delay convergence loci. Firstly, convergence was highly dependent on the delay-update step size ν. This is not abnormal for gradient descent algorithms, but in our case was sensitive to the point where essentially a different step-size was required for every initial delay condition.

Moreover, it is known that since speech is highly periodic, the algorithm can become trapped in local maxima which exhibit strong correlation due to periodicity, rather than to delay [19]. A study of this effect, such as may be found in [1], shows that the joint entropy surface of two linearly mixed speech signals has multiple maxima, as functions of delay. These extrema are very closely spaced – approximately 10 samples apart. Thus one must obtain exceptionally precise estimates for the delays apriori, as initial conditions, to within ± 5 samples. Even with this knowledge, an un-tuned step-size can destroy convergence.

Computation of a cross-correlation sequence can provide an estimate of delays; for 2 mixtures, calculating a cross-covariance between mixtures and taking the indices of the two highest values would seem to produce good delay estimates. The technique may also be readily extended for a greater number of sources. However, we found that cross-covariance estimates were just as prone to the same trapping along oscillatory (voice-like) segments as the gradient rule was. The ultimate fact to be faced is that periodic speech portions behave much too like delayed versions. Thus no local algorithm, processing sample by sample, can distinguish between these two scenarios.

In our view, there is not much hope for adapting the delays separately in the above manner. Including the delays as a part of the deconvolving filter may be inefficient but permits additional freedom in terms of extra coefficients, allowing for increased robustness.

3.3 Convolved Mixtures

One difficulty with evaluating blind separation algorithms more complex than the instantaneous case is in the choice of an error measure. For feedback systems the question is especially difficult given the recursive nature of their equations. The best we may do is to compare the weights to some ideal solution.

From Eqs. 10 and 11, the reader should verify that for two sources and two mixtures, an ideal solution is obtained via:

$$W_{12} = -A_{12}A_{22}^{-1},\ W_{21} = -A_{21}A_{11}^{-1}$$

This suggests the two performance indices:

$$P_1(z) = -W_{12}A_{22}A_{12}^{-1},$$
$$P_2(z) = -W_{21}A_{11}A_{21}^{-1}$$

where each $P(t)$ ideally is a scaled impulse response. The simplest such measure is:

$$e = \sum_i \left(\sum_n \frac{|P_i(n)|}{\max_k |P_i(k)|} - 1 \right) \qquad \text{Eq. 19}$$

Because our reference performance is an ideal case, we cannot expect $e \to 0$ over time. The error measure is thus *relative,* but still useful for observing the dynamic behaviour of filter coefficients.

For feedforward systems, a performance filter matrix $\mathbf{P}(z) = \mathbf{W}(z) \cdot \mathbf{A}(z)$ provides a clue. Again, consider the case of two sources and two mixtures. Ignoring permutation by manual reordering, perfect separation *and* dereverberation occurs if $\mathbf{P} = \begin{bmatrix} P_{11}(t) & P_{12}(t) \\ P_{21}(t) & P_{22}(t) \end{bmatrix}$ possesses impulse responses on diagonals and zero responses on off-diagonals, i.e. $P_{11}(t) = P_{22}(t) = \delta(t),\ P_{12}(t) = P_{21}(t) = 0.$ This would be too much to expect, however, given filtering indeterminacies.

An approach is to compute the total energy of diagonal terms and compare to the energy contained in off-diagonal terms. This measure is flawed though, since greater signal distortion (a significant impulse response) is perceived as greater separation. We thus adopt the following error metric, which uses the ratio of *maximum* filter values in each performance row:

$$e_1 = \frac{\max_t |P_{12}(t)|}{\max_t |P_{11}(t)|},\ e_2 = \frac{\max_t |P_{21}(t)|}{\max_t |P_{22}(t)|}$$

The attenuation of unwanted source component for each channel may then be defined as a decibel representation of the "signal-to-unwanted signal" ratio:

$$\mathrm{A}_1 = -20\log_{10} e_1$$
$$\mathrm{A}_2 = -20\log_{10} e_2$$

Tests of the algorithms are performed for two voices, with two sets of mixing scenarios:

Set I:

$$\mathbf{A}_{11}(z) = 1 - 0.4z^{-25} + 0.2z^{-45},$$
$$\mathbf{A}_{12}(z) = 0.4z^{-20} - 0.2z^{-28} + 0.1z^{-36}$$
$$\mathbf{A}_{21}(z) = 0.5z^{-10} + 0.3z^{-22} + 0.1z^{-34}$$
$$\mathbf{A}_{22}(z) = 1 - 0.3z^{-20} + 0.2z^{-38}$$

Set II:

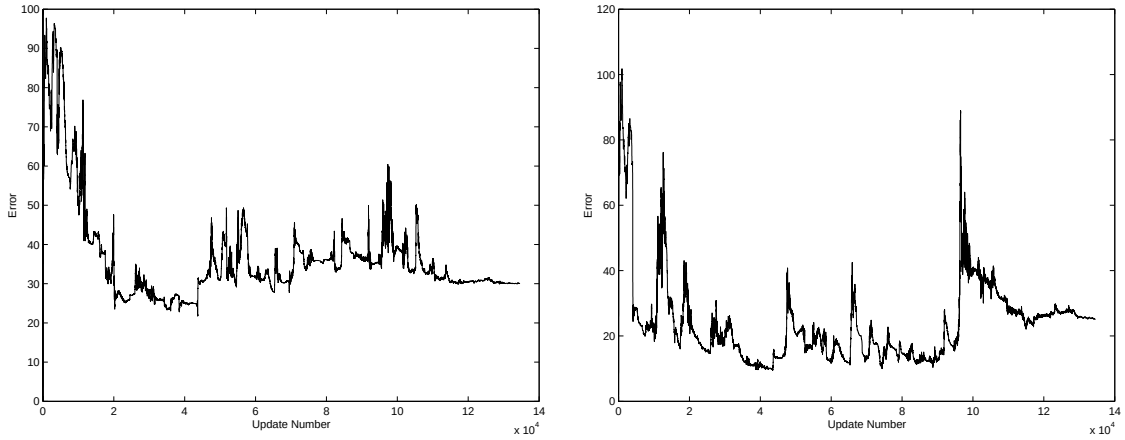
$$\mathbf{A}_{11}(z) = 1 + 0.8z^{-1} + 0.7z^{-2} + 0.4z^{-3} + 0.3z^{-4} + 0.25z^{-5} + 0.2z^{-6} + 0.15z^{-7}$$
$$\mathbf{A}_{12}(z) = 0.6 + 0.5z^{-1} + 0.5z^{-2} + 0.4z^{-3} + 0.3z^{-4} + 0.2z^{-5} + 0.25z^{-6} + 0.1z^{-7}$$
$$\mathbf{A}_{21}(z) = 0.5 + 0.5z^{-1} + 0.4z^{-2} + 0.35z^{-3} + 0.3z^{-4} + 0.3z^{-5} + 0.2z^{-6} + 0.1z^{-7}$$
$$\mathbf{A}_{22}(z) = 1 + 0.9z^{-1} + 0.8z^{-2} + 0.6z^{-3} + 0.4z^{-4} + 0.35z^{-5} + 0.3z^{-6} + 0.15z^{-7}$$

Of course both sets are completely artificial, but the principles and operation of the procedures are most easily demonstrated with relatively simple filters. The first matrix filter

involves far-spaced coefficients, with a relative delay of 10 and 20 samples. The second test set uses 8 closely spaced taps.

3.3.1 Feedback Architecture

We test Torkkola's deconvolution algorithm (Eq. 12 & 13) on the test cases I and II. FIR unmixing filters w_{ij} of length $L = 100$ were utilised, initialised to $w_{ijk} = \delta_{ij}(k)$, step size $\mu = 0.001$, and a hyperbolic tangent nonlinearity. Updates occurred at every speech sample. The error curves as defined by Eq. 19 are given below:



(a) Test Set I (b) Test Set II

Fig 21: Error curves for the separation of two mixed voices in the Infomax feedback architecture

The method performs reasonably well for Test Set I; and seemingly well for Test Set II. A listening test shows some deception: the output exhibits good separation (though not perfect) for the first set of mixings, but terrible separation in the latter case. There is little to no filtering distortion of the original sources for case 1, however – in correspondence with expected theory.

3.3.2 Feedforward Architecture

Since the block update format of Amari's Natural Gradient Algorithm is far more efficient and stable, we only consider implementation of the procedure given in Section 2.3.3.

Parameters were set as follows: FIR filters of length L=128 (implying blocks of length 256, 50% overlap), initialized to $w_{ijk} = \delta_{ij}(k)$, step-size $\mu = 0.0005$ and a nonlinearity of $-40\,\mathrm{sgn}(\mathbf{u})$. All 9 seconds of data were passed through the algorithm a number of times until the filters iterated to the point of convergence.

The attenuation curves A_1 and A_2 are produced below, as well as the impulse responses of the resultant performance matrix:

Test Set I:

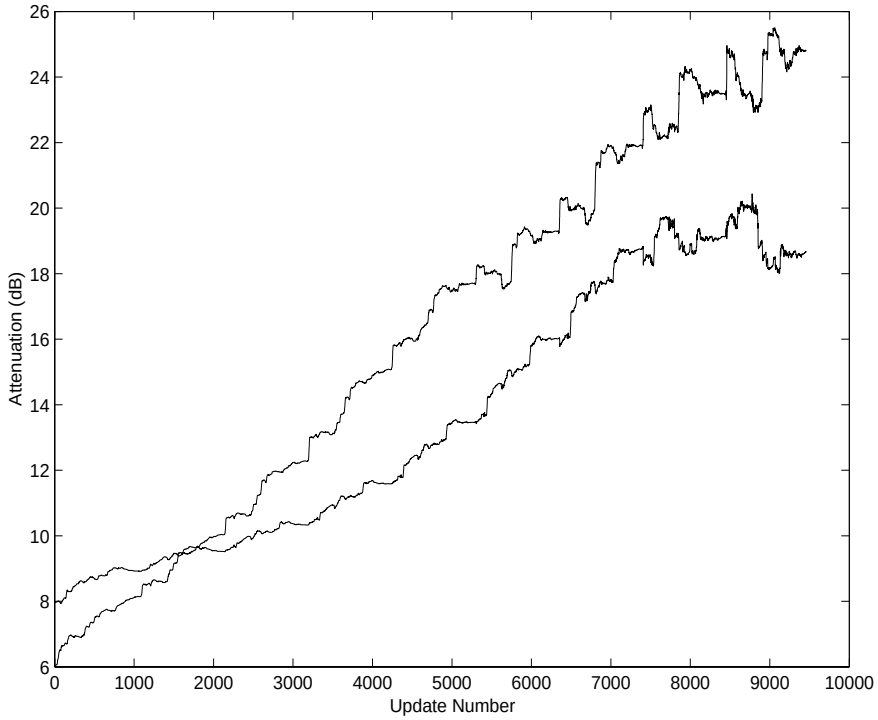
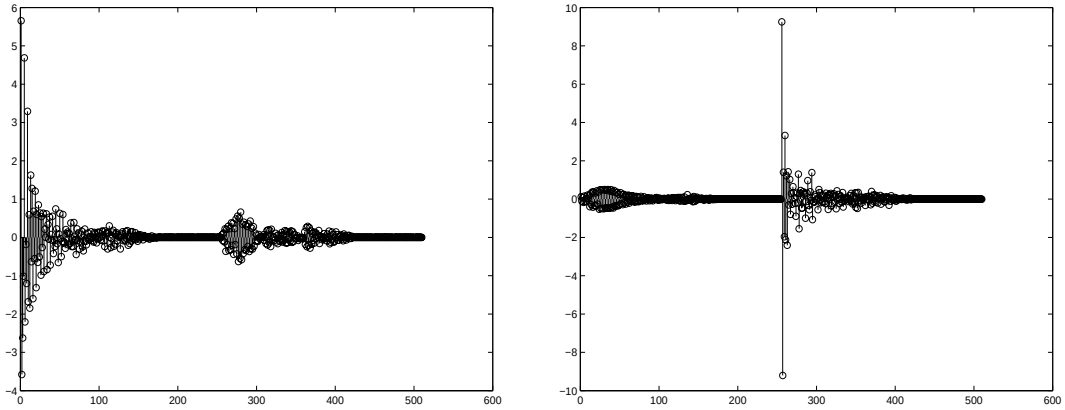


Fig 22: Channel attenuations of unwanted signals



(a) Performance row 1 (b) Performance row 2

Fig 23: Final impulse responses of the filters in **P**(z). Each row contains two filters, the impulse responses which are concatenated.

Test Set II:

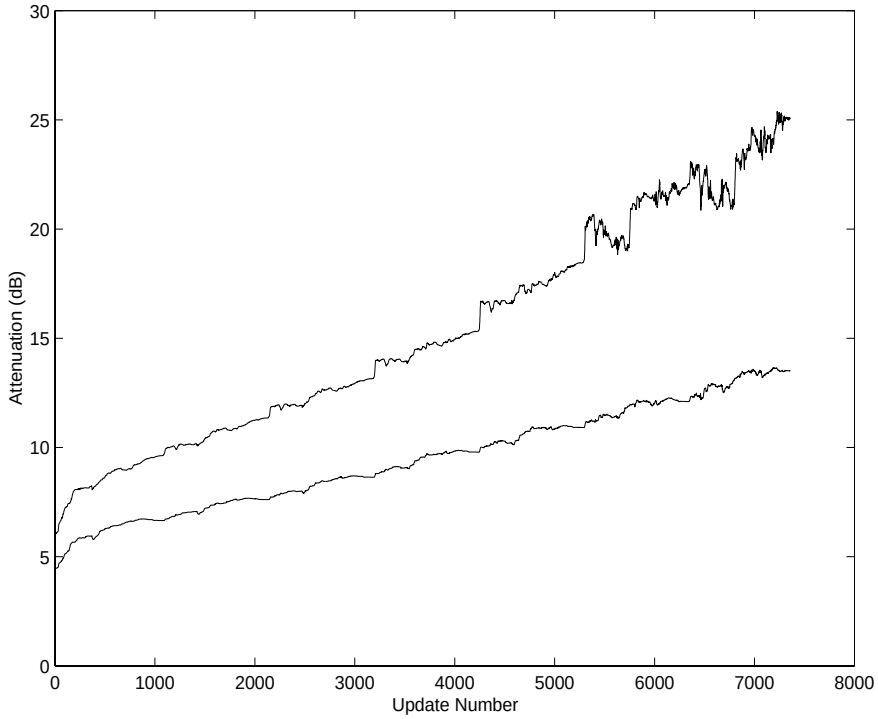
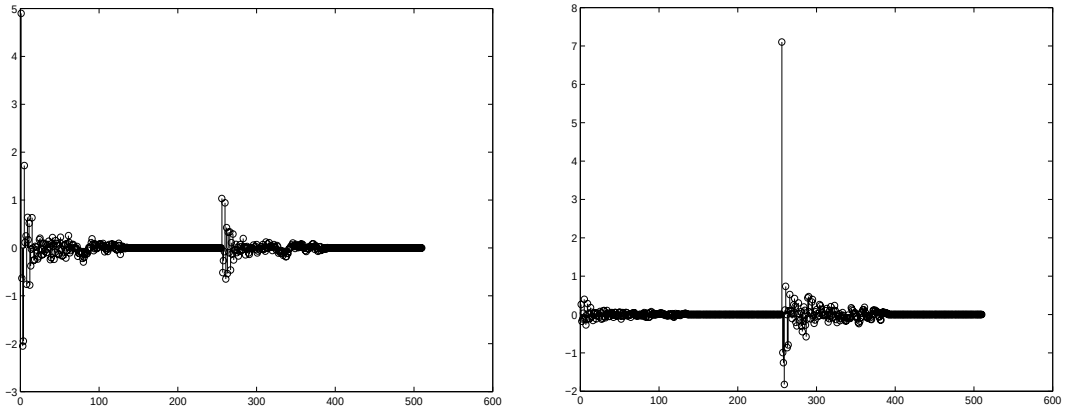


Fig 24: Channel attenuations of unwanted signals

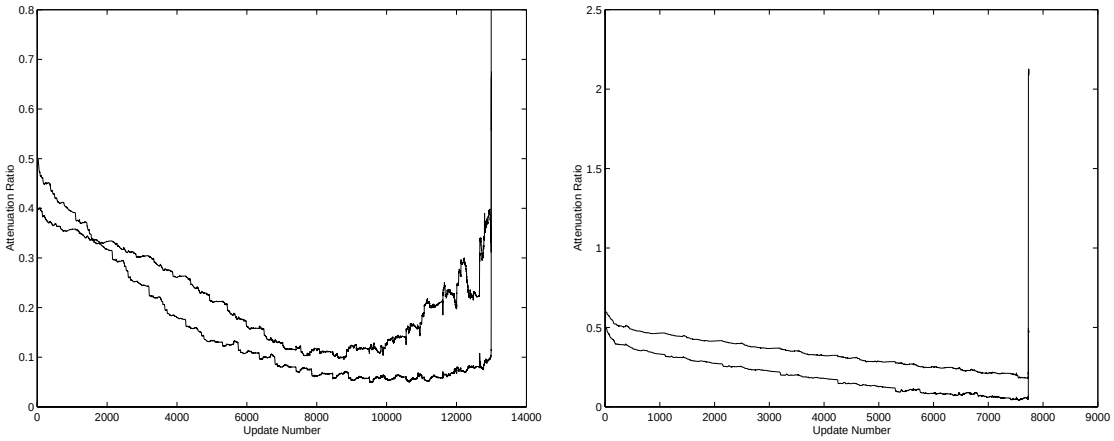


(a) Performance row 1 (b) Performance row 2

Fig 25: Final impulse responses of the filters in **P**(z). Each row contains two filters, the impulse responses which are concatenated.

An intuitive sense of the level of separation can be found by examining the final performance rows. As in Section 3.3.1, the subjective auditory impression is of considerable attenuation of the unwanted component, but not perfect separation. Notice in each channel, the initial strong impulse tap is followed by negative taps. This corresponds with temporal decorrelation – an insidious side-effect of all feedforward designs. The whitening effect can be plainly discerned in the output audio waveforms as well – speech sounds less full and more metallic.

A serious problem manifests itself once the natural gradient algorithm seems to converge. From Figs. 22 & 24 it might appear that further iteration results in superior separation. This is not the case. In fact the algorithm becomes unstable and diverges! Observe the following two graphs of the attenuation ratios e_1 & e_2, after continued update:



(a) Test Case I (b) Test Case II

Fig 26: Channel attenuation ratios for the feedforward architecture.

Alteration of the step-size (by decreasing), or (increasing) filter lengths help "stave off" divergence, but not indefinitely – for all paths, the algorithm eventually diverges. The update can be terminated just prior to instability but then the adaptive property is destroyed. An underlying mechanism behind this divergence may have to do with a whitening pursuit direction overtaking the separation pursuit direction. The next section addresses this difficulty.

3.4 LP Residual-Domain Weight Update

We now attack the problem of whitening, implementing the residual-domain filter update expounded in Section 2.5. As a first step, illustrating at least the plausibility of ICA upon speech residuals, we present an application of the algorithm upon two instantaneously mixed sources:

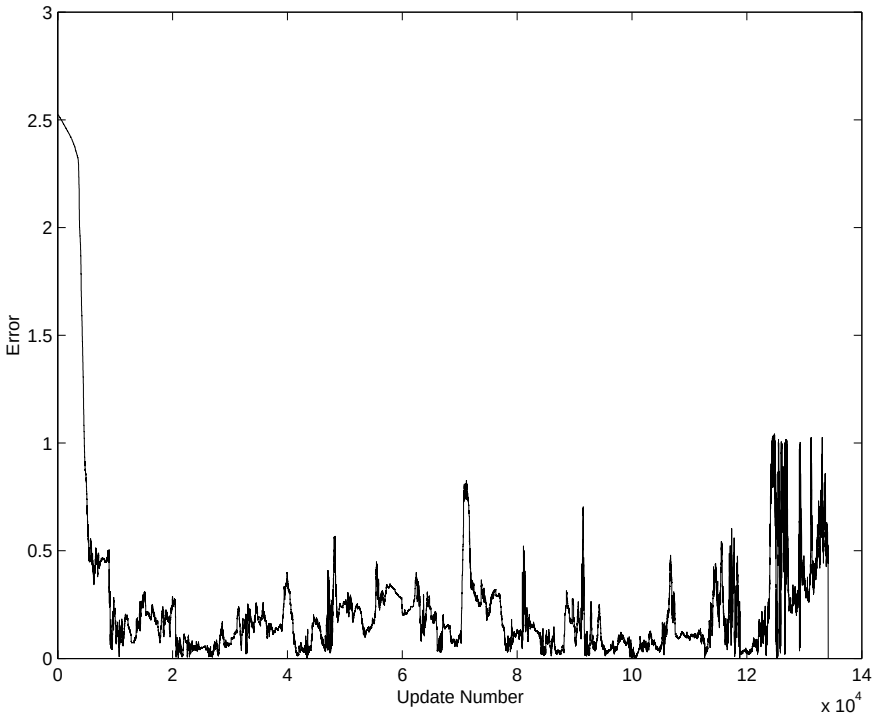


Fig 27: Error curve for residual-domain weight update for the instantaneous case.

An LP analysis filter of order 4 was used, and the nonlinearity $-120\,\mathrm{sgn}(\mathbf{u})$. Note that the signum amplitude coefficient (120) has been increased drastically from the usual one in Section 3.2.2 to account for the smaller variance of LP residuals as compared to regular speech. The associated audio files show perfect separation, as does the above error curve.

We move on to the convolutional case. With all parameters identical to those of the basic feedforward case of Section 3.3.2, we apply LP pre-processing of order 4 with a window size of 512 samples before ICA. The attenuation curves and final performance matrix filters are shown below:

Test Set I:

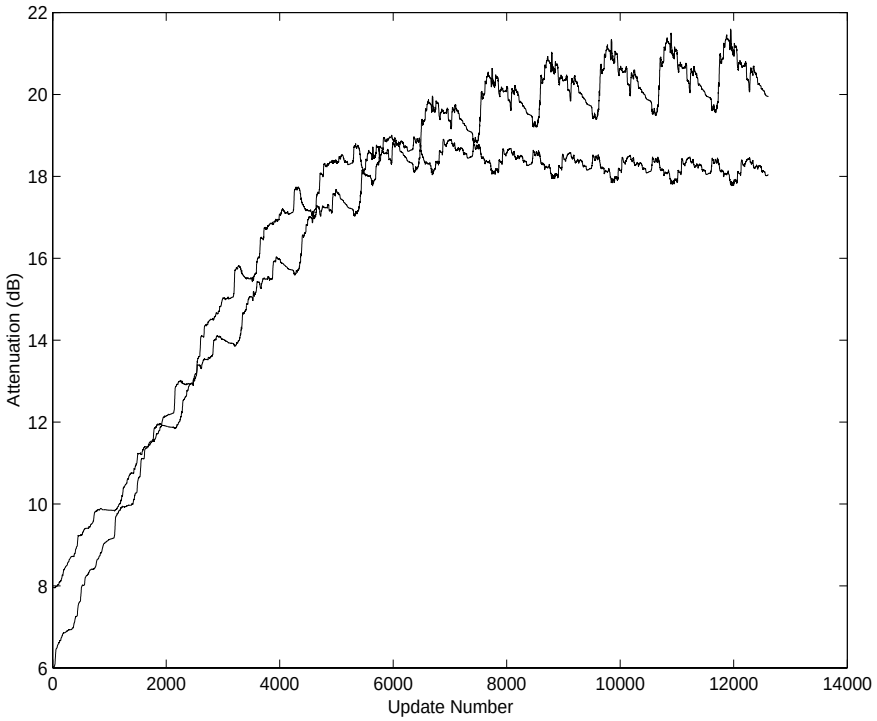


Fig 28: Channel attenuations of unwanted signals

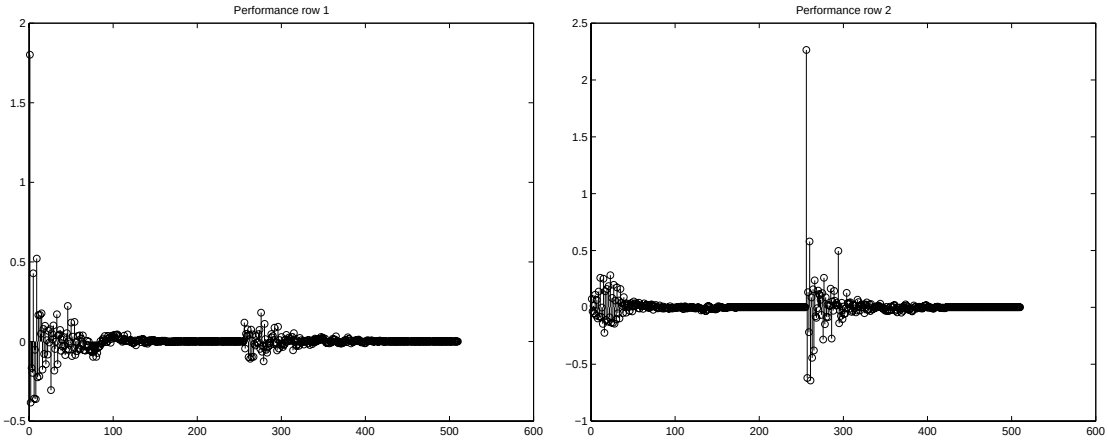


Fig 29: Final impulse responses of the filters in **P**(z)

Test Set II:

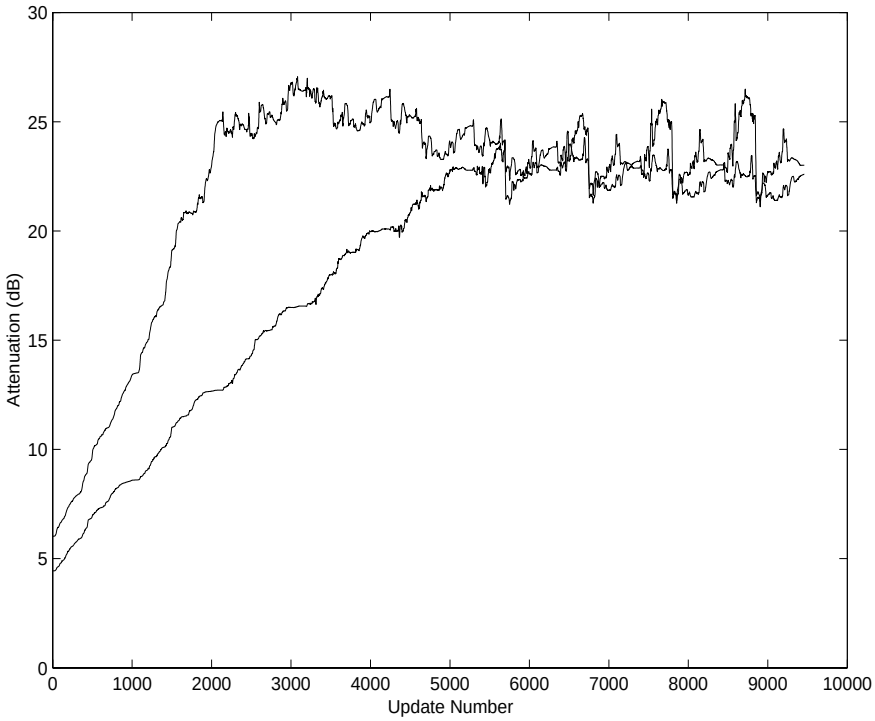


Fig 30: Channel attenuations of unwanted signals

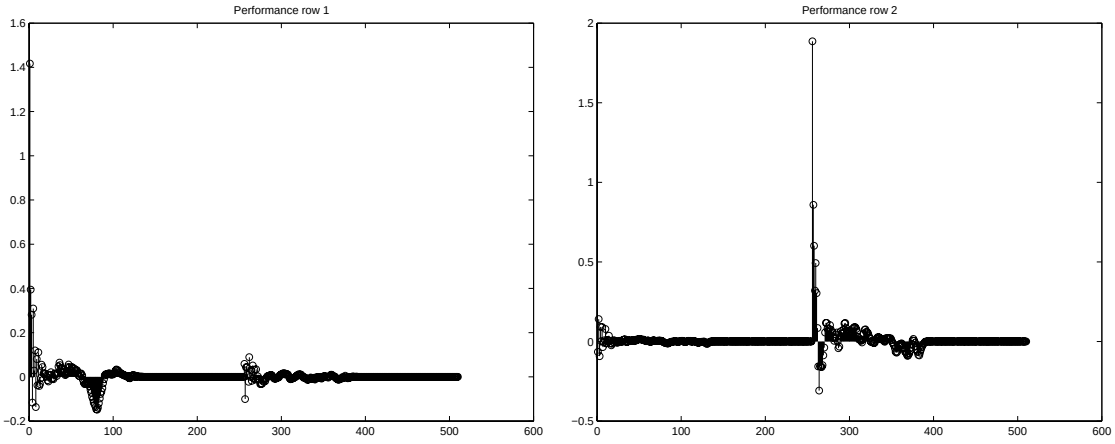


Fig 31: Final impulse responses of the filters in **P**(z)

Important considerations may be taken from these graphs:

1) The LP residual-domain update improves convergence drastically, obtaining the maximum in less than half the time required for that of the basic update.

2) Vastly increased stability: the algorithm does not diverge after finding the extremum.

3) Whitening of sources is reduced significantly

The third point may be witnessed by first-hand observation of the output speech waveforms, which sound far more like the original sources than the versions produced in Section 3.3.2. Separation quality itself is also superior to that of the basic algorithm. A more objective indication of whitening can be obtained from the autocorrelation sequence of the output waveforms (or more accurately the autocorrelation sequence of the original sources filtered by the diagonal elements of the performance matrix). The autocorrelation sequence of a segment of normal speech is shown below:

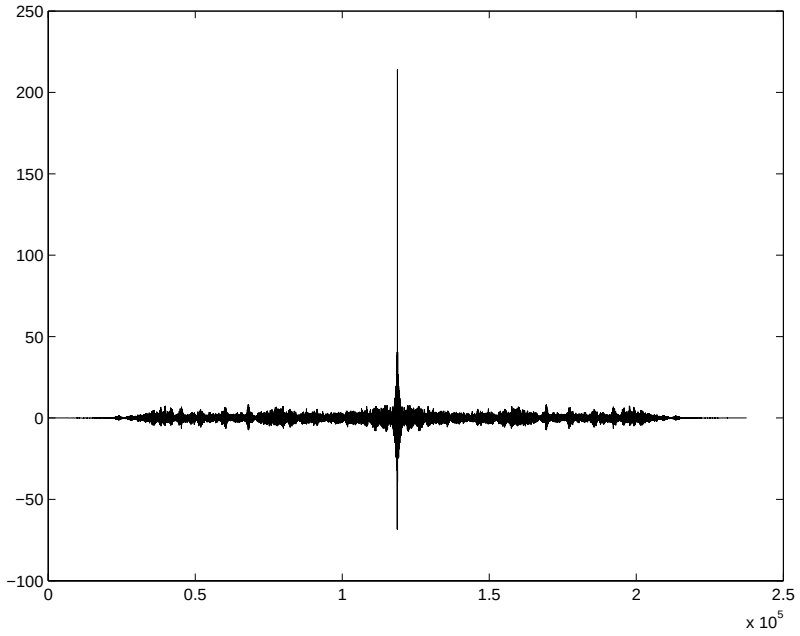


Fig 32: Autocorrelation function of normal speech

A relative measure of whitening (which we call the correlation factor) may then be defined by calculating the energy of the function within some neighbourhood (e.g. ±250 samples) of the central peak. High energy indicates strong correlation, or little whitening, and low energy indicates weak correlation, or heavy whitening.

The following table gives indications of the degree of whitening produced by the basic and residual-domain algorithms:

	Original Speech	Test Set I		Test Set II	
		Basic Update	Residual Update	Basic Update	Residual Update
Speaker 1	65.36	20.59	39.72	38.09	42.83
Speaker 2	51.22	23.23	42.40	50.46	51.54

Table 1: Correlation factors for the block feedforward algorithm and residual-domain filter update algorithm

These observations provide objective confirmation that whitening is reduced when using the Residual Update algorithm.

Finally, it may be of some peripheral interest to display the computation time required for each algorithm:

Algorithm	Time for 1 update (s)	Number of updates for convergence	Total time to convergence (s)
Feedback	0.0067	20000	134
Basic Feedforward	0.0068	8000	54.4
Residual Update	0.0068	4500	30.6

Table 2: Processing times for Blind Deconvolution Algorithms

4 Conclusion

We have presented a number of variants on Independent Component Analysis, and performed comparative experimental tests. Overall, the information maximization cost function seems to be superior to methods relying on *n*th-order statistics, as well as being especially suited for adaptive stochastic filtering. Separate adaptation of delays has been shown to be non-robust and difficult. In the choice between feedback and feedforward architectures, a feedforward architecture is more general for being able to learn non minimum-phase systems, but exhibits the whitening effect for temporally correlated sources. The LP residual-update introduced allows one to use the feedforward architecture while eliminating whitening, as well as providing greater stability and faster convergence.

Appendix A: Timeline of Topics

May 1st – 14th:	Background reading on Information Theory, Probability, and DSP. Implementation of the basic instantaneous mixing problem and the "Infomax" solution.
May 15th – 31st:	Investigation of the effects of dynamic mixing in Blind Signal Separation. Shift to convolutive mixing problem.
June 1st – 14th:	The pure delay problem. Implementation of an adaptive delay system.
June 14th – 21st:	Optimization techniques for faster convergence.
June 21st – July 14th:	Implementation of feedback and feedforward InfoMax deconvolution algorithms.
July 14th – July 21st:	Incomplete and overcomplete representations; nonlinearity parameter matching.
July 21st – Aug 1st:	Time-delayed and multiple decorrelation methods.
Aug 1st – Aug 21st:	Blind Signal Separation on LP residuals.
Aug 21st – Sept 10th:	Report writing.

Appendix B: Audio Guide

The CD Audio Files contains the results of all simulations discussed in this report. It is organized into the following directories:

Sources

Section 3.1.1, Section 3.1.2, Section 3.1.3, Section 3.1.4

Section 3.2

Section 3.3.1, Section 3.3.2

Section 3.4

The directory “sources” contains four wav files which form the original speech material for all simulations.

Each “Section” directory corresponds with the appropriate section in the Experimental Results chapter, and contains files of two types: mix*n*.wav, and unmix*n*.wav (or demix*n*.wav), where *n* is a number. The “mix” files are the sensor channels (received microphone signals), and the “unmix” files are the output channels after algorithm application. All sound files have been amplitude scaled to the interval [-1, 1].

References

[1] T. Lee, *Independent Component Analysis: Theory and Applications*. Kluwer Academic Publishers, 1998.

[2] D. Chan, *Blind Signal Separation*. Ph. D. Thesis, Cambridge University, 1997.

[3] M. Solazzi, F. Piazza and A. Uncini, "Nonlinear Blind Source Separation by Spline Neural Networks". *Proc. ICASSP*, May 2001.

[4] L. Parra and C. Spence, "Convolutive Blind Source Separation based on Multiple Decorrelation". *Proc. Neural Networks Signal Processing*, Sept. 1998.

[5] H. Attias, "Independent Factor Analysis". *Neural Computation* 11(4): 803-852.

[6] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[7] C. G. Putonet, A. Prieto, C. Jutten, M. Rodriguez-Alvarez and J. Ortega, "Separation of sources: A geometry-based procedure for reconstruction of n-valued signals". *Signal Processing*, 46:267-284, 1995.

[8] W. Pereira, *Modifying LPC Parameter Dynamics to Improve Speech Coder Efficiency*. Masters Thesis, McGill University, 2001.

[9] D. O'Shaughnessy, *Speech Communications: Human and Machine*. IEEE Press, 2000.

[10] N. Murata, S. Ikeda and A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals". *Technical Report BSIS-98-2, BSI, RIKEN*, 1998.

[11] S. Haykin, *Adaptive Filter Theory, 3rd Edition*. Prentice Hall, 1996.

[12] J. Cardoso, "High-order Contrasts for Independent Component Analysis". *Neural Computation*, 11(1):157-192, 1999.

[13] P. Comon, “Independent Component Analysis – a New Concept?”. *Signal Processing*, 36(3): 287-314.

[14] J. Cardoso, “A Tetradic Decomposition of 4^{th}-order Tensors. Application to the Source Separation Problem”. *Algorithms, Architectures & Applications, Vol 3 of SVD & Signal Processing*: 375-382, 1995.

[15] E. Moreau, “An Any-Order Generalization of JADE for Complex Source Signals”. *Proc. ICASSP*, May 2001.

[16] A. Bell and T. Sejnowski, “An Information-Maximization Approach to Blind Separation and Blind Deconvolution”. *Neural Computation*, 7(6):1129-1159, 1995.

[17] S. Amari, “Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10: 251-276, 1998.

[18] L. Molgedey and H. Schuster, “Separation of a Mixture of Independent Signals Using Time Delayed Correlations”. *Physical Review Letters* 72(23): 3634-3637.

[19] K. Torkkola, “Blind Separation of Delayed Sources Based on Information Maximization”. *Proc. ICASSP*, May 1996.

[20] R. H. Lambert, *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*. Ph. D. Thesis, University of Southern California, 1996.

[21] P. Smaragdis, “Blind Separation of Convolved Mixtures in the Frequency Domain”. *Intern. Workshop on Independence & Artificial Neural Networks*, Feb. 1998.

[22] G. Georgiou, “Activation functions for Neural Networks in the Complex Domain”. *First Intern. Conference on Fuzzy Theory and Technology*, Oct. 1992.

[23] K. Torkkola, “Blind Separation of Convolved Sources Based on Information Maximization”, *IEEE Workshop on Neural Networks for Signal Processing*, 1996.

[24] S. Amari, S. Douglas, A. Cichocki and H. Yang, "Novel On-line Adaptive Learning Algorithms for Blind Deconvolution Using the Natural Gradient Approach". *Proc. 11th IFAC Symp. On System Identification*, July 1997.

[25] K. Na, S. Kang, K. Lee and S. Chae, "Frequency-Domain Implementation of Block Adaptive Filters for ICA-Based Multichannel Blind Deconvolution", *Proc. ICASSP*, May 1999.

[26] K. Rahbar and J. Reilly, "Blind Source Separation of Convolved Sources by Joint Approximate Diagonalization of Cross-Spectral Density Matrices". *Proc. ICASSP*, May 2001.

[27] J. Cardoso and B. Laheld, "Equivariant Adaptive Source Separation". *IEEE Transactions on Signal Processing*, 45(2): 434-444, 1996.

[28] A. Hyvarinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis". *IEEE Transactions on Neural Networks* 10(3): 626-634, 1999.

[29] J. Cardoso, "Infomax and Maximum Likelihood for Blind Source Separation". *IEEE Signal Processing Letters*, 4(4): 112-114.

[30] N. Charkani and Y. Deville, "Optimization of the Asymptotic Performance of Time-Domain Convolutive Source Separation Algorithms". *European Symposium on Artificial Neural Networks*: 273-278, April 1997

[31] B. Yegnanarayana, P. Satyanarayana Murthy, C. Avendano and H. Hermansky, "Enhancement of Reverberant Speech using LP Residual". *Proc. ICASSP*, May 1998.

[32] B. Gillespie, H. Malvar and D. Florencio, "Speech Dereverberation via Maximum-Kurtosis Subband Adaptive Filtering", *Proc. ICASSP*, May 2001.

[33] M. Joho, H. Mathis and R. Lambert, "Overdetermined Blind Source Separation: Using More Sensors than Source Signals in a Noisy Mixture". *Proc. Independent Component Analysis & Blind Signal Separation*, June 2000.