THE COHERENCE FUNCTION IN BLIND SOURCE SEPARATION OF CONVOLUTIVE MIXTURES OF NON-STATIONARY SIGNALS

Craig L. Fancourt and Lucas Parra
Adaptive Image & Signal Processing Group, Sarnoff Corp.
201 Washington Rd., Princeton, NJ 08543
cfancourt@sarnoff.com, lparra@sarnoff.com

ABSTRACT

We propose a new performance criteria and update mechanism for the blind decorrelation of an array of sensor measurements into independent sources, assuming each sensor measures a different convolutive mixture of statistically independent non-stationary sources. Specifically, the criteria is the sum of the magnitude squared coherence functions between all possible distinct pairs of outputs produced by a matrix of adaptable filters operating on the sensor measurements in the frequency domain. We then derive an efficient overlap-save online update equation based on stochastic gradient descent and recursive estimation of the coherence functions. We demonstrate separation within fractions of a second and convergence within a few seconds on real room recordings. We attribute this speed to the normalization and recursive estimates of the coherence functions.

1 INTRODUCTION

While the theoretical underpinnings of the blind source separation (BSS) problem have advanced tremendously in the last decade (see [1]-[2]), the development of fast, efficient, and robust algorithms that can solve real-world problems is still lacking. This is particularly true for problems involving the presence of many sources, more sources than sensors, the interference of diffuse noise, and the separation of convolutive mixtures. It is the latter problem with which are concerned in this paper.

The problem of separating convolutive mixtures of unknown sources arises in several application domains, of which the most famous is the so-called cocktail party problem. There, the problem is to recover the speech of multiple speakers who are simultaneously conversing in a room, where their acoustic speech signals are each filtered by a different speaker-to-microphone room response, depending on their position, and then linearly mixed at the microphones. In the special case of the

meeting room transcription task, the goal is to recover the individual speakers from the microphone signals sufficiently well to allow for use in an automatic speech recognizer.

1.1 Approach

We have previously studied this problem for the special case when the source signals are non-stationary processes ([3]-[5]). Non-stationarity can arise through changes in the first-order distribution of a signal, as evidenced by changes in power, changes in second-order joint distributions across time, as evidenced by changes in spectrum, or higher-order changes.

Source separation is primarily based on the assumption of statistical independence of the source signals. For stationary signals, second-order statistics (decorrelation) is not sufficient to identify and invert the mixing coefficients [7], and higher-order statistics have to be considered either explicitly ([9]-[11]) or implicitly ([14]-[17]). However, for non-stationary signals, varying second-order statistics provides a sufficient constraint for separation [6]. In this case, multiple covariance matrices estimated at different times can be simultaneously diagonalized ([4],[6]). This approach has been demonstrated for the convolutive case ([3]-[5],[18]), and has been thoroughly analyzed for the instantaneous case.

Thus, in attempting to separate non-stationary signals using multiple decorrelation, we are faced with the problem of designing an algorithm that *requires* non-stationary signals for convergence. However, in adaptive signal processing, we are used to formulating solutions to problems that depend on stationarity, and then applying them to non-stationary signals on the assumption of "adiabatic" changes, where the rate-of-change of stationarity is less than the time constant of the adaptation. For example, in developing the recursive least squares (RLS) algorithm, the optimal solution of the Wiener-Hopf equations are solved at time t in terms of the optimal solution at time t-1, given the assumption of stationarity. However, this does not prevent us from applying the RLS to non-stationary signals, through the introduction of an appropriate forgetting factor.

If we were to attempt an analogous approach for BSS, we might ask the question: given that we have found a set of filters that "best" separate the outputs at time t-1, and a new set of measurements at time t, how do we update the filters to best separate the outputs, making use of the solution at time t-1? An answer to this question is not so simple. The reason for this is that what we do with the new measurement depends on whether it is part of the previous stationary regime, or represents a transition to a new stationary regime. In the first case, the new data should be used to improve the estimate of the current covariance, implying the use of a large forgetting factor. In the second case, the data represents the beginning of new covariance matrix for simultaneous diagonalization with previous covariance matrices, implying a small forgetting factor is appropriate. Therefore, in addition to the conventional trade-off between convergence speed and misadjustment, we now have a trade-off between estimation accuracy and novel information when measuring correlation.

As a compromise, the approach we take here is to focus on how to effectively and efficiently measure decorrelation, and then turn that measure into a criteria for adaptation purposes. In this paper, we propose to use the *coherence function* as a measure of signal decorrelation. The coherence function is the frequency-domain equivalent of the correlation coefficient, and represents the degree of correlation as a function of frequency. It has the desirable property of being scaled such that it is independent of the absolute power of either signal.

2 PROBLEM STATEMENT

The problem we seek to solve is the following: N unknown source signals are convolutively mixed and measured by M sensors

$$\mathbf{x}(t) = \mathbf{A}*\mathbf{s}(t) \tag{1}$$

where $\mathbf{s}$ is an unknown (Nx1) vector of source signals, $\mathbf{A}$ is an unknown (MxN) mixing matrix of channel impulse responses, and $\mathbf{x}$ is a measured (Mx1) vector. The convolution operator * here implies both matrix multiplication and convolution. We then seek a matrix of filters operating on the sensor measurements

$$\mathbf{y}(t) = \mathbf{W}*\mathbf{x}(t) \tag{2}$$

such that the components of the (Nx1) output $\mathbf{y}$ are statistically independent, where $\mathbf{W}$ is a (NxM) matrix of filter impulse responses.

In order to understand what an independence criteria can accomplish, it suffices to determine the set of all operations on $\mathbf{s}$ such that the resulting signals are still independent. Clearly a reordering of the components of $\mathbf{s}$ does not affect their independence. The components of $\mathbf{s}$ can also be separately filtered, either linearly or nonlinearly, without affecting their independence. Thus, $\mathbf{y}$ can only approximate $\mathbf{s}$ to within a permutation and filtering operation. The latter limitation means that BSS is distinct from the problem of blind deconvolution. That is, independence based BSS by itself cannot recover the components of $\mathbf{s}$ from filtered versions of themselves. For this reason, the diagonal components of $\mathbf{W}$ in the time domain are often fixed to a unity gain delta function, possibly with a delay.

In the time domain, independence must be tested not only at the same instant of time, but for all possible combinations of delays of the components of $\mathbf{y}$. This problem can be ameliorated by performing the separation in the frequency domain. In the frequency domain, convolution becomes multiplication and (2) becomes

$$\mathbf{Y}(\omega, t) = \mathbf{W}(\omega, t) \cdot \mathbf{X}(\omega, t) \tag{3}$$

Note that because the signals are *assumed* non-stationary, we have written their frequency response as an *implicit* function of time. We have written the (NxM) matrix of filter frequency responses, $\mathbf{W}(\omega,t)$, as an implicit function of time with an eye towards adaptation rules that we will develop later.

Equations (2) and (3) describe *any* linear system. Ultimately, we must implement them in a specific architecture. In this paper, we use finite impulse response (FIR) filters because this allows the actual filtering operation to be carried out in the frequency domain.

3 THE COHERENCE FUNCTION AS SEPARATION CRITERIA

The criteria we adopt is the sum of the *magnitude squared coherence functions* between all *N*x(*N*-1)/2 possible distinct pairs of outputs

$$J = \sum_t \sum_{i,j} \left| C_{Y_i Y_j}(\omega, t) \right|^2 \tag{4}$$

where $C_{Y_i Y_j}(\omega, t)$ is the *coherence function* between outputs *i* and *j*, defined by

$$C_{Y_i Y_j}(\omega, t) = \frac{S_{Y_i Y_j}(\omega, t)}{\sqrt{S_{Y_i Y_i}(\omega, t) S_{Y_j Y_j}(\omega, t)}} \tag{5}$$

and where $S_{Y_i Y_j}(\omega, t)$ is the *cross-power spectral density* between outputs *i* and *j* at time *t*. The squared coherence function is real, constrained to lie between 0 and 1 for all frequencies, and is identically equal to one when *i=j*. This latter property means that it is immaterial whether the summation in (4) includes the case *i=j*.

We can express all these equations in matrix form as

$$J = \sum_t \left\| \mathbf{C}_{YY}(\omega, t) \right\|^2 = \sum_t \text{trace}[\mathbf{C}_{YY}^H(\omega, t) \cdot \mathbf{C}_{YY}(\omega, t)] \tag{6}$$

where $\mathbf{C}_{YY}$ is a (*N*x*N*) matrix of coherence functions whose components are $C_{Y_i Y_j}$. Equation (6) represents the *Frobenius squared norm* of the coherence function matrix. Again, because the diagonal elements of the coherence function matrix are identically one, it is immaterial whether we include them in the criteria. We can also express the matrix of coherence functions

$$\mathbf{C}_{YY}(\omega, t) = \Lambda_{YY}^{-1/2}(\omega, t) \cdot \mathbf{S}_{YY}(\omega, t) \cdot \Lambda_{YY}^{-1/2}(\omega, t) \tag{7}$$

in terms of a (*N*x*N*) matrix of cross-power spectral densities between the outputs, $\mathbf{S}_{YY}$, whose components are $S_{Y_i Y_j}$, and a diagonal matrix, Λ_{YY}, whose diagonal elements are $S_{Y_i Y_i}$. Inserting (7) into (6) results in:

$$J = \sum_t \text{trace}[\Lambda_{YY}^{-1}(\omega, t) \cdot \mathbf{S}_{YY}(\omega, t) \cdot \Lambda_{YY}^{-1}(\omega, t) \cdot \mathbf{S}_{YY}(\omega, t)] \tag{8}$$

3.1 Estimating the output-output cross-power spectral densities

Formally, the cross-power spectral density is the Fourier transform of the expected value of the cross-correlation in the time domain. However, it can also be obtained as the expected value of the product of the signals in the frequency domain. In order to efficiently estimate the output-output cross-power spectral density, we use a recursive estimator

$$\mathbf{S}_{YY}(\omega, t) = \gamma \ \mathbf{S}_{YY}(\omega, t - T) + (1 - \gamma) \ \mathbf{Y}(\omega, t) \cdot \mathbf{Y}^{H}(\omega, t) \tag{9}$$

where γ is a forgetting factor, constrained to $0 < \gamma < 1$ for stability, and T is a block processing time (frame rate) that represents the time it takes to estimate $\mathbf{Y}$. The forgetting factor and block processing time combine to make the effective memory of the estimator to be $T/(1-\gamma)$. Taking the expected value of both sides of (9) readily shows that it is an *unbiased* estimator for *stationary* signals.

3.2 Weight update

Clearly, in order to capture short-term non-stationarity we must use the stochastic gradient approximation. Thus, in order to find the weight-update equation, we take the derivative of the criteria (8) with respect to the complex weights in the frequency domain and then drop the summation over time, updating the weights at the end of each time block.

We are then faced with the question of whether to take the derivatives with respect to the power spectral densities $S_{Y_i Y_i}$. Since they always appear in the denominator, this implies simultaneously decorrelating the outputs *and* maximizing the output power. However, the output powers are already constrained by fixing the diagonal filters of $\mathbf{W}$ to unity gain delta functions. Hence, for the purposes of this paper, we regard them as a constant normalization factor. Then, it is not difficult to show that the gradient update equation is

$$\Delta\mathbf{W}(\omega, t) = -\eta \Lambda_{YY}^{-1}(\omega, t) \cdot [\mathbf{S}_{YY}(\omega, t) - \Lambda_{YY}(\omega, t)] \cdot \Lambda_{YY}^{-1}(\omega, t) \cdot \mathbf{S}_{YX}(\omega, t) \tag{10}$$

where $\mathbf{S}_{YX}$ is a (NxM) matrix of cross-power spectral densities between the *outputs* and the *inputs*:

$$\mathbf{S}_{YX}(\omega, t) = \gamma \ \mathbf{S}_{YX}(\omega, t - T) + (1 - \gamma) \ \mathbf{Y}(\omega, t) \cdot \mathbf{X}^{H}(\omega, t) \tag{11}$$

It is important to note that the recursive nature of the cross-power spectral density estimates in (9) and (11) means that $\mathbf{S}_{YY}$ cannot be obtained directly from $\mathbf{S}_{YX}$ through a simple multiplication by the weights. This differs from all previous methods that we have seen, including our previous work, which use a single recursive estimate of $\mathbf{S}_{XX}$ involving the inputs only, and then obtains $\mathbf{S}_{YX}$ and $\mathbf{S}_{YY}$ through multiplications involving the weight matrix $\mathbf{W}$.

3.3 Implementation details

The entire algorithm consists of equations (3) and (9)-(11) and is entirely compatible with the overlap-save method of frequency domain adaptive filtering. The overlap-save method implements linear convolution in the frequency domain with the discrete Fourier transform (DFT), or its efficient counterpart, the fast Fourier transform (FFT). However, since the DFT corresponds to circular convolution in the time domain, the filters must be padded with zeros, in turn requiring the use of a larger input buffer. As a result, only the latter part of the output in the time domain is valid. In the context of the present algorithm, it is thus incorrect to directly use the complex output $\mathbf{Y}=\mathbf{W}*\mathbf{X}$ in updating the cross-power spectral densities in (9) and (11). Rather, they must first be transformed into the time domain (also required to obtain the system output), and the invalid parts zeroed prior to transforming back into the frequency domain to obtain a valid $\mathbf{Y}$ for use in (9) and (11). Note that this is not required for $\mathbf{X}$, since the input buffer is always filled with valid input samples prior to transforming into the frequency domain. Although other frame rates relative to the filter size can be used, a 50% overlap is the most compuationally efficient and is the one adopted for the simulations presented next.

The computational complexity of the algorithm scales linearly in the number of inputs and quadratically in the number of outputs. For the simulation to be presented next, a two input - two output problem at a sampling rate of 8 kHz with 256 taps ran in approximately 1/20 real-time for compiled c-code on a 866 MHz Pentium III. Thus, the algorithm is entirely suitable for real-time operation for many-input, many-output problems.

4 EXPERIMENTS

The data set used here first appeared in [3], and later [4]-[5]. Two live speakers were recorded in a real room of dimensions 3m x 3.6m x 2.3m using two unidirectional microphones, 50 cm apart and 150 cm from the speakers, and sampled at 8 kHz. The training data is a 15 second recording where both speakers are continuously and simultaneously talking. The test data consists of another 15 second recording where the two speakers alternately say the digits such that only one speaker is active at a time. The two data sets were recorded consecutively to ensure that the speakers maintained their position and thus that the room responses would not change.

For the test data, the active periods of each speaker were hand segmented in order to obtain an accurate measurement of signal separation. Whenever a performance measure was required during training, the training weights were used to filter the entire 15 seconds of test data. The resulting output was then analyzed using the aforementioned segmentation such that whenever a speaker was talking, the power in both the enhanced and rejection channels were measured and accumulated. The *signal to interference ratio* (SIR) was then calculated as

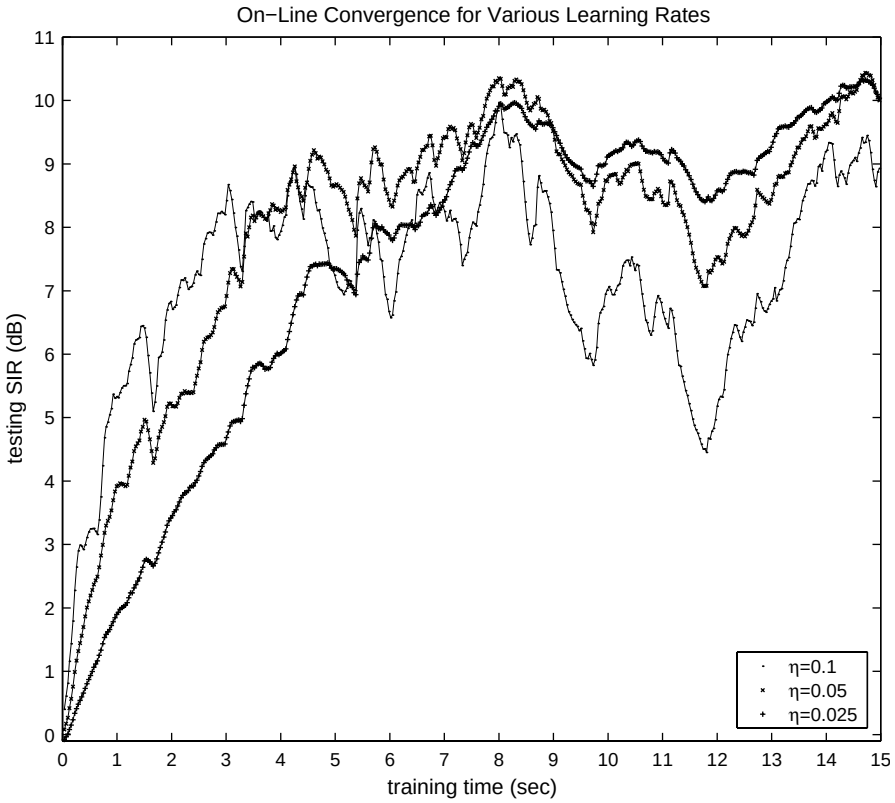

Figure 1. Test set performance during on-line adaptation on training set for various learning rates. Other parameters are $\gamma = 0.5$ and $N_{taps} = 256$.

$$SIR = 10 \cdot \log_{10}\left[\frac{P_{enhanced}}{P_{rejection}}\right] \tag{12}$$

The effect of various learning rates on the algorithm's convergence is shown in Fig. 1. The test set performance was measured during *on-line* adaptation on the training set after *each* weight update, which for 256 taps at 8 kHz occurred every 0.032 s. With the forgetting factor set at γ=0.5, the effective memory depth was thus 0.064 s. At the highest learning rate (0.1), a separation of 3 dB was achieved after only 10 weight updates or approximately 1/3 s, and a separation of 6 dB was achieved in approximately 1.3 s. However, this learning rate also exhibited some undesirable misadjustment about the mean, particularly around the 12 s mark. It is not clear what causes this temporary degradation in performance, but its extended nature suggests possible movement of one of the speakers during the training utterances. The smaller learning rates are less susceptible to this, at the expense of a slower initial convergence time.

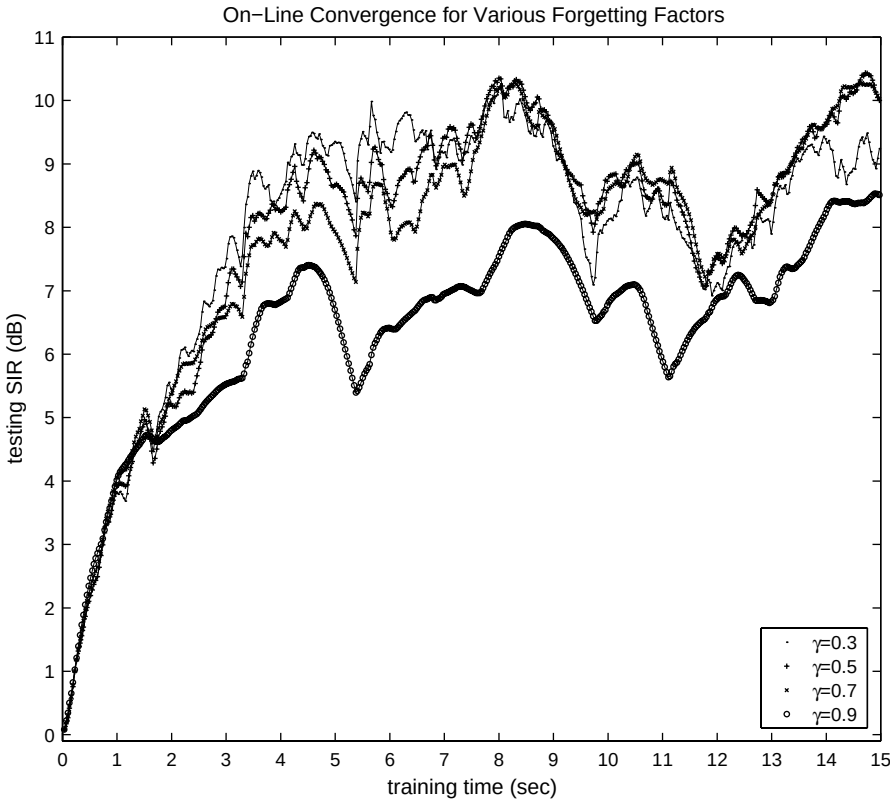

Figure 2. Test set performance during on-line adaptation on training set for various forgetting factors. Other parameters are $\eta = 0.05$ and $N_{\text{taps}} = 256$.

The effect of various forgetting factors on the algorithm's on-line convergence is shown in Fig. 2. Early in the training (< 2s), the forgetting factor does not seem to play an important role. However, beyond this, a forgetting factor of γ=0.9 clearly under performs the other settings. At the other extreme, a forgetting factor of γ=0.3 outperforms the other settings between 3s and 7s, but then under performs. Overall, the best performance was obtained for a forgetting factor of γ=0.5.

Finally, the *off-line* performance for various filter sizes is shown in Fig. 3 as a function of the length of training data. The algorithm was trained for 10 iterations on increasing lengths of training data, using the weights from the previous length as a seed. Performance was then measured on the test data using the weights at the end of the final training iteration. For 512 taps, a separation of over 11 dB was obtained using approximately 1 s of data. Thus, very little data is needed to achieve good separation. For 1024 taps, a separation of better than 12 dB is consistently maintained, but it takes 5 s of training data to achieve this level.

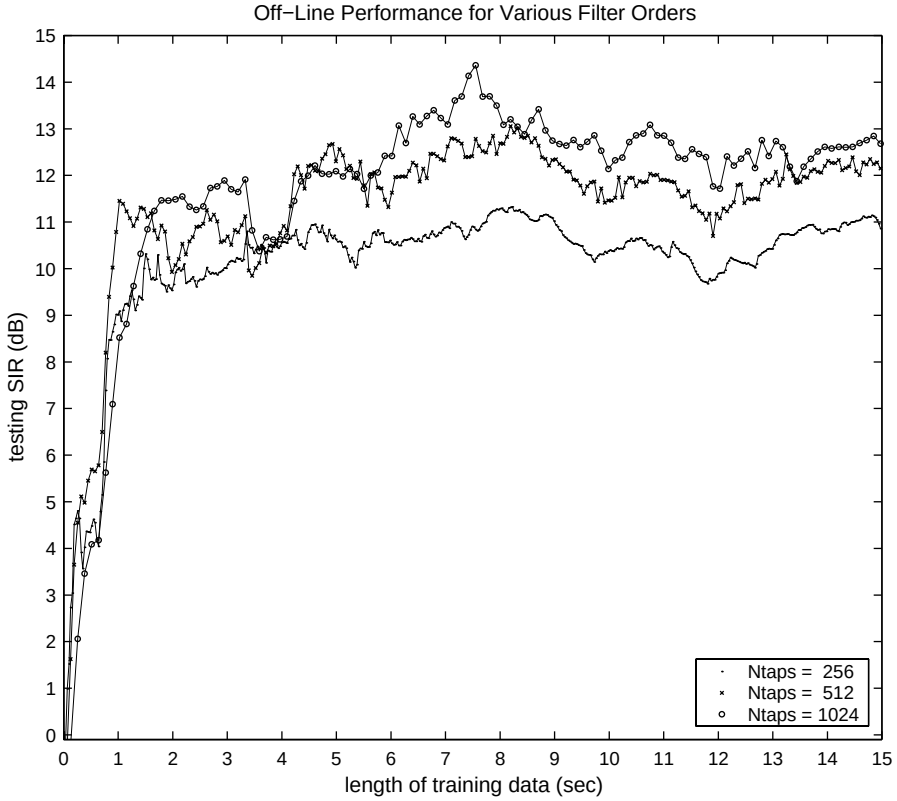

Figure 3. Test set performance after off-line training as a function of training data length for various filter lengths. Other parameters are $\eta = 0.025$ and $\gamma = 0.5$.

5 CONCLUSIONS

The present work is an extension of our previous work on the blind decorrelation of non-stationary signals. However, it differs significantly in two important respects: the use of the coherence function and its corresponding normalization, and the independent estimates of the output-output and input-output cross power spectral densities. The combination of these two improvements results in very fast convergence. However, because the BSS literature typically reports steady-state rather than convergence performance, and due to a lack of standardized data sets, we cannot claim that the algorithm is faster than all others. Nevertheless, the low computational complexity and fast convergence clearly shows that it is suitable for real-time operation. In addition, the potential remains for improving performance through adaptation of the learning rate and/or forgetting factor, particularly as the latter parameter relates to the rate-of-change of stationarity. We also plan to study the permutation and scaling problem as it relates to the details of the algorithm.

6 REFERENCES

[1] S. Haykin (ed.), *Unsupervised adaptive filtering, vol. 1: blind source separation*, John Wiley & Sons, New York, 2000.

[2] S. Roberts and R. Everson (eds.), *Independent component analysis: principles and practice*, Cambridge Univ. Press, Cambridge, U.K., 2001.

[3] L. Parra, C. Spence, and B. De Vries, Convolutive blind source separation based on multiple decorrelation, in *IEEE Workshop on Neural Networks for Signal Processing*, Cambridge, UK, 1998.

[4] L. Parra and C. Spence, Convolutive blind separation of non-stationary sources, *IEEE Trans. Speech Audio Proc.*, vol. 8, no. 3, pp. 320-327, 2000.

[5] L. Parra and C. Spence, On-line convolutive blind source separation of non-stationary signals, *Journal of VLSI Signal Processing*, vol. 26, pp. 39-46, 2000.

[6] E. Weinstein, M. Feder, and A.V. Oppenheim, Multi-channel signal separation by decorrelation, *IEEE Trans. Speech Audio Processing*, vol. 1, no. 4, pp. 405-413, 1993.

[7] S. Van Gerven and D. Van Compernolle, Signal separation by symmetric adaptive decorrelation: stability, convergence, and uniqueness, *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1602-1612, 1995.

[8] M. Kawamoto, A method of blind separation for convolved non-stationary signals, *Neurocomputing*, vol. 22, no. 1-3, pp. 157-171, 1998.

[9] D. Yellin and E. Weinstein, Multichannel signal separation: methods and analysis, *IEEE Trans. Signal Processing*, vol. 44, no. 1, pp. 106-118, 1996.

[10] H.-L. N. Thi and C. Jutten, Blind source separation for convolutive mixtures, *Signal Processing*, vol. 45, no. 2, pp. 209-229, 1995.

[11] V. Capdevielle, C. Serviere, and J.L. Lacoume, Blind separation of wide-band sources in the frequency domain, in *Proc. ICASSP 95*, pp. 2080-2083, 1995.

[12] S. Shamsunder and G. Giannakis, Multichannel blind signal separation and reconstruction, *IEEE Trans. Speech Audio Processing*, vol. 5, no. 6, pp. 515-528, 1997.

[13] S. Cruses and L. Castedo, A Gauss-Newton method for blind separation of convolutive mixtures, in *Proc. ICASSP 98*, Seattle, WA, 1998.

[14] R. Lambert and A. Bell, Blind separation of multiple speakers in a multipath environment, in *Proc. ICASSP 97*, pp. 423-426, 1997.

[15] S. Amari, S.C. Douglas, A. Cichocki, and A.A. Yang, Multichannel blind deconvolution using the natural gradient, in *Proc. 1st IEEE Workshop on Signal Processing App. Wireless Comm.*, pp. 101-104, 1997.

[16] T. Lee, A. Bell, and R. Lambert, Blind separation of delayed and convolved sources, in *Proc. Neural Information Processing Systems* 96, 1997.

[17] P. Smaragdis, Blind separation of convolved mixtures in the frequency domain, *Neurocomputing*, vol. 22, pp. 21-34, 1998.

[18] J. Principe, Simultaneous diagonalization in the frequency domain (SDIF) for source separation, in *ICA 99*, pp. 245-250, 1999.

[19] K.-C Yen and Y. Zhao, Adaptive Co-channel speech separation and recognition, *IEEE Trans. Signal Processing*, vol. 7, no. 2, 1999.

[20] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 1996.