GEOMETRIC OPTIMIZATION METHODS FOR BLIND SOURCE SEPARATION OF SIGNALS

Kamran Rahbar and James P. Reilly

Electrical & Computer Eng.
Mcmaster University, Hamilton, Ontario, Canada
Email: kamran@reverb.crl.mcmaster.ca, reillyj@mcmaster.ca

ABSTRACT

In this paper we develop a new blind signal separation (BSS) algorithm using conjugate gradient optimization over the Stiefel manifold. We express the BSS problem mathematically as an optimization problem with an orthonormal constraint. This can be expressed as an *unconstrained* optimization over the Stiefel manifold [1]. To derive the algorithm, we only use second order statistics of the observed signals a criterion which has been shown to be sufficient for separation providing that sources have linearly independent temporal correlations. The new optimization method displays a quadratic convergence property. Simulation results corresponding to two different optimization strategies are presented that verify the performance of the new algorithm and also it's convergence behaviour.

1. INTRODUCTION

In a blind source separation (BSS) problem, the objective is to separate independent sources that are mixed through an unknown mixing environment where no information is available about the sources or the environment. A simple BSS scenario is when the mixing environment modelled as a matrix of scalars, referred to as instantaneous mixing. So far, many methods have been proposed to solve the BSS problem for instantaneous mixture case. Some methods use information measures or higher order statistics as a criterion for separation (e.g., [2], [3], [4]) while few others rely only on second order statistics by exploiting the temporal information of signals (e.g.,[5], [6], [7]) No matter what kind of criterion is used, a blind source separation problem most often ends up with an optimization task where one needs to minimize (or maximize) a cost function in order to achieve separation. As proposed by some methods (see e.g., [3] and [2]) the BSS problem can be reduced to a simpler form by performing a spatial pre-whitening. In this case the problem is simplified to finding a unitary matrix V that separates the outputs. In order to find the separating network (V) we need to maximize (or minimize) a cost function subject to an orthogonality constraint ($V^TV = I$). So far the methods that have been proposed in literature consider V as a multiplication of Jacobian matrices and solve the problem by optimizing the cost function with respect to rotation angles of each Jacobian matrix. In general orthogonal constraints can be represented geometrically by Grassman and Stiefel manifolds. In [1] new algorithms for optimization on the these manifold have been discussed. In this paper we have developed two new BSS algorithms which minimize the second order cross moments between outputs subject to the above-mentioned unitary constraint. We have developed unconstrained gradient descent and conjugate gradient methods over the Stiefel manifold as in [1] to achieve the separation. The results for the two algorithms are compared on the basis of simulations. Good separation performance is achieved for both algorithms, with quadratic convergence displayed for the conjugate gradient-based method.

2. PROBLEM FORMULATION AND SEPARATION CRITERION

2.1. Preliminary

Assume that we have source signal vector $\mathbf{s}(m)$ consists of n sources $\mathbf{s}(m) = (s_1(m), s_2(m), \cdots, s_n(m))^T$ where the source signals are real, zero mean, stationary processes. Also we make the further assumptions that the sources $s_i(m)$ are mutually uncorrelated and they have different spectral contents. Consider the observation $\mathbf{x}(m)$ as the linear, instantaneous mixture of source signals given by:

$$\mathbf{x}(m) = A\mathbf{s}(m). \tag{2.1}$$

The BSS problem is: given only $\mathbf{x}(m)$, find a demixing matrix B such that the output vector given by

$$\mathbf{y}(m) = B\mathbf{x}(m), \tag{2.2}$$

is within a scaled and permuted version of the original source vector. Since the exchange of a fixed scalar factor between a given source and corresponding column of A does not change the observation, without any loss of generality we can assume that the sources have unit power. Based on this assumption the covariance matrix of observed signal $x(m)$ can be written as:

$$R_x = E[\mathbf{x}(m)\mathbf{x}^T(m)] = AE[\mathbf{s}(m)\mathbf{s}^T(m)]A^T = AA^T. \tag{2.3}$$

Here we have used the assumption that $E[\mathbf{s}(m)\mathbf{s}^T(m)] = I$. Replacing A with its singular value decomposition: $A = U\Sigma V^T$ in equation (2.3) we have:

$$R_x = U\Sigma^2U^T. \tag{2.4}$$

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC), Centre for Information Technology Ontario (CITO)

From the eigendecomposition of matrix R_x we notice that the mixing matrix A can be identified up to a unitary matrix. To see this more clearly we define a whitening matrix W as:

$$W = \Sigma^{-1} U^T. \tag{2.5}$$

By applying W to the input observation we obtain:

$$\mathbf{z}(m) = W\mathbf{x}(m) = V^T \mathbf{s}(m). \tag{2.6}$$

As can be seen from (2.6) the BSS problem has been simplified to finding the unitary matrix V. To identify V based on the assumptions already made on the sources we can use a second order statistics (SOS) approach.

2.2. Cost Function

As is shown in [8] V can be identified by diagonalizing the spatial whitened covariance matrix $R_z(l)$ defined as:

$$R_z(l) = E[\mathbf{z}(m)\mathbf{z}^T(m-l)] = VE[\mathbf{s}(m)\mathbf{s}(m-l)]V^T, \tag{2.7}$$

given that for any sources $s_i(m)$ and $s_j(m)$ there is a $l > 0$ such that:

$$E(s_i(m)s_i(m-l)) \neq E(s_j(m)s_j(m-l)). \tag{2.8}$$

The problem with the above approach is that we need to find a priori a time lag l such that the above condition is satisfied. This becomes especially important when the sources have close spectrums. Another approach as suggested by [5] is to simultaneously (jointly) diagonalize a set of L whitened covariance matrices $S = \{R_z(l) | l = 1, \cdots, L\}$. In their method they use an extension of the Jacobi technique [9] for the joint approximate diagonalization of the set of covariance matrices mentioned above. Notice that joint diagonalizing of the set of covariance matrices S can be interpreted as minimizing the norm of off-diagonal elements of $R_y(k)$ given as:

$$R_y(l) = V^T R_z(l) V, \tag{2.9}$$

for all $1 \leq k \leq L$. In other words, joint diagonalization is equivalent to the solution of the following optimization problem

$$\begin{array}{rcl} min \quad \Gamma_1(V) & = & \sum_{l=1}^{L} \sum_{i \neq j} r_{ij}^2(l) \\ subject\ \ to \quad V^T V & = & I \end{array} \tag{2.10}$$

where V is a n-by-n orthonormal matrix and r_{ij} are the off-diagonal elements of $R_y(l)$ and are given by

$$r_{ij}(l) = E[y_i(m)y_j(m-l)], \tag{2.11}$$

and $y_i(m)$ is the ith element of the output vector $\mathbf{y}$ given as:

$$\mathbf{y}(m) = V\mathbf{z}(m). \tag{2.12}$$

Since the Frobenius norm of the covariance matrix $R_y(k)$ is invariant with respect to V, minimizing the norm of off-diagonal terms of $R_y(k)$ is equivalent to maximizing the norm of diagonal terms. In other words instead of (2.11) we can write:

$$\begin{array}{rcl} max \quad \Gamma_2(V) & = & \sum_{k=1}^{L} \sum_{i} r_{ii}^2(l), \\ subject\ \ to \quad V^T V & = & I. \end{array} \tag{2.13}$$

Notice that

$$\sum_i r_{ii}^2(l) = \frac{1}{2}((Tr(R_y(l)))^2 + \sum_{i \neq j}(r_{ii}(l) - r_{jj}(l))^2), \tag{2.14}$$

and $Tr(R_y(l))$ is invariant with respect to V. Hence for maximizing (2.14), we only need to maximize the second term (or minimizing it's negative) of equation (2.14) with respect to V or :

$$\begin{array}{rcl} min \ \ \Gamma(V) & = & -\frac{1}{2} \sum_{l=1}^{L} \sum_{i \neq j} (r_{ii}(l) - r_{jj}(l))^2 \\ subject\ to \ \ V^T V & = & I. \end{array} \tag{2.15}$$

As it can be seen from (2.16) we have an optimization problem with an orthonormal constraint. In [1] the writers provide a framework for solving problems that involve such constraints. Based on their work in next section we discuss the optimization methods to solve (2.16).

3. OPTIMIZATION METHODS

In this section we introduce optimization methods for solving (2.16). As mentioned before the orthonormal constraint $V^T V = I$ can be represented by a nonlinear space known as Stiefel manifold. Using differential geometry ideas, the constrained optimization problem on given by (2.16) can be considered as an unconstrained one on the Stiefel manifold.

3.1. Gradient descent on Stiefel manifold

In linear Euclidean space we have the following update rule for smooth unconstrained optimization of a objective function $f(X)$ given as:

$$X_k = X_{k-1} + tH_{k-1}, \tag{3.16}$$

where H is the search direction calculated based on the knowledge of gradient or Hessian of the objective function and t is the step size parameter typically chosen using line search methods. Similar concepts can be carried over to optimization on a manifold, by translating the operations mentioned above to the suitable ones on the manifold. Notice that the update rule in equation (3.16) is done on a line while on a manifold this update should be done on a geodesic, where by it's definition is the curve of shortest length between two points on a manifold. As is shown in [1], on the Stiefel manifold the equation for the geodesic emanating from V_{k-1} in direction of H_{k-1} ,where V_{k-1} and H_{k-1} are n-by-p matrices such that $V_{k-1}^T V_{k-1} = I_p$ and $A = V_{k-1}^T H_{k-1}$ is skew-symmetric, is given by:

$$V_k = V_{k-1}E(t) + QF(t). \tag{3.17}$$

Here $E(t)$ and $F(t)$ are p-by-p matrices given by matrix exponential:

$$\left(\begin{array}{c} E(t) \\ F(t) \end{array}\right) = exp\left[t\left(\begin{array}{cc} A & -R^T \\ R & 0 \end{array}\right)\left(\begin{array}{c} I_p \\ 0 \end{array}\right)\right] \tag{3.18}$$

Where I_p is a p-by-p identity matrix and Q and R are the QR decomposition of:

$$QR = (I - V_{k-1}V_{k-1}^T)H. \tag{3.19}$$

For gradient descent we set $H_k = -G_k$ where G_k is the gradient of cost function $\Gamma(V_k)$ on the Stiefel manifold and is given by

$$G_k = \Gamma_{V_k} - V_k\Gamma_{V_k}^T V_k, \tag{3.20}$$

where Γ_{V_k} is the n-by-p matrix of partial derivatives of $\Gamma(V_k)$ with respect to elements of V_k

$$(\Gamma_{V_k})_{ij} = \frac{\partial\Gamma(V_k)}{\partial v_{ij}^k}, \tag{3.21}$$

where v_{ij}^k is the (i,j)th element of the V_k. For steepest descent, equations (3.17) and (3.20) are all we need to minimize the cost function $\Gamma(V_k)$ on the Stiefel manifold. At each iteration k we first find G_k using equation (3.20), then we set $H_k = -G_k$ and using equation (3.17) we find the update value for V_{k+1}. To choose the step size t we have different options: either we can choose a constant value for t or we can use a successive step size reduction method such as Armijo Rule described in [10].

3.2. Conjugate Gradient on the Stiefel Manifold

In optimization over a linear space, another method that is faster than steepest descent but still only needs gradient information of the objective function is the conjugate gradient method [10]. The search direction (H_k) in the conjugate gradient method at each step is calculated using a linear combination of the gradient of the cost function (G_k) at current step and the search direction at the previous step.

$$H_k = -G_k + \beta_k H_{k-1}, \tag{3.22}$$

Where β_k is given by:

$$\beta_k = \frac{G_k^T G_k}{G_{k-1}^T G_{k-1}}. \tag{3.23}$$

As is shown in [10], by using the conjugate gradient for quadratic problems, a solution can be attained after finite number of steps. For non quadratic functions the convergence may not happen after certain number of steps but the method still provides good convergence properties. To perform conjugate gradient on a manifold we need to know how to parallel transport a tangent vector from one point of the manifold to another point. Notice that on the Stiefel manifold both G_k and H_{k-1} are tangent vectors and the new search direction is found by adding the gradient vector to the parallel transported version of previous search direction vector. In Euclidean space, we move vectors in parallel by moving the base of the arrow. On an embedded manifold in Euclidean space, if we use the same concept to move a tangent vector the result won't necessarily be a tangent vector. For parallel transport of tangent vectors on a manifold we parallel transport the vector in infinitesimal steps as we do in Euclidean space and then in each step we remove the normal component of the transferred vector such that the remaining is still tangent to the manifold. On the Stiefel manifold we can find the parallel transported tangent vector H_k from point V_k to V_{k+1} from

$$\tau H_k = H_k E(t) - V_k R^T F(t), \tag{3.24}$$

where $E(t)$ and $F(t)$ are obtained using equation (3.18) and R is obtained form (3.19). The equation for updating the search direction in conjugate gradient method over the Stiefel manifold is given as:

$$H_{k+1} = -G_{k+1} + \beta_k\tau H_k, \tag{3.25}$$

where

$$\beta_k = \frac{\langle G_{k+1}, G_{k+1}\rangle}{\langle G_k, G_k\rangle}, \tag{3.26}$$

and $\langle\Delta_1, \Delta_2\rangle$ represents the inner product between two tangent vectors on the Stiefel manifold and is defined as:

$$\langle\Delta_1, \Delta_2\rangle = Tr(\Delta_1^T(I - \frac{1}{2}V_k V_k^T)\Delta_2). \tag{3.27}$$

The equation above is commonly known as Fletcher-Reeves formula. Another equation for β is the Polak-Ribiere formula given as:

$$\beta_k = \frac{\langle G_{k+1} - G_k, G_{k+1}\rangle}{\langle G_k, G_k\rangle}. \tag{3.28}$$

4. ALGORITHM

In this section we derive a new algorithm for blind source separation of instantaneous mixtures based on the criterion introduced in (2.16) and the optimization methods that were discussed in the previous section. Because we want to use steepest descent and conjugate gradient over the Stiefel manifold we need to calculate the gradient of the objective function over the manifold. To do so we first find Γ_{V_k}, the matrix of the partial derivatives of $\Gamma(V_k)$ with respect to elements of V_k. We have:

$$(\Gamma_{V_k})_{ij} = \frac{\partial\Gamma_{V_k}}{\partial v_{ij}} = -\sum_{l=1}^{L}\frac{\partial r_{ij}(l)}{\partial v_{ij}}\sum_{\substack{p=1 \\ p\neq i}}^{n}(r_{ii}(l) - r_{pp}(l)), \tag{4.29}$$

and

$$\sum_{\substack{p=1 \\ p\neq i}}^{n}(r_{ii}(l) - r_{pp}(l)) = nr_{ij}(l) - Tr(R_y(l)), \tag{4.30}$$

where $R_y(l) = E[\mathbf{y}(m)\mathbf{y}(m-l)^T]$. Notice that here since V_k is an unitary matrix we have:

$$Tr(R_y(l)) = Tr(R_z(l)) = c \quad \forall l, \tag{4.31}$$

where c is a constant and independent of V_k. We also find that:

$$\frac{\partial r_{ij}(l)}{\partial v_{ij}} = (2V_k R_z(l))_{ij} \tag{4.32}$$

Substituting above equations in 4.29 we can show the following for $\Gamma(V_k)$:

$$\Gamma_{V_k} = -2\sum_{l=1}^{L}\Lambda_y(l)V_k R_z(l), \tag{4.33}$$

where

$$\Lambda_y(l) = nD_y(l) - cI_n, \tag{4.34}$$

and $D_y(l)$ represents a diagonal matrix whose diagonal elements are the same as diagonal elements of $R_y(l)$ and I_n is an n-by-n identity matrix. By inserting Γ_{V_k} in (3.20) we can calculate the gradient of our objective function over the Stiefel manifold. After doing some algebraic manipulations we will obtain:

$$G_k = 2\sum_{l=1}^{L}(\Lambda_y(l)R_y(l) - R_y(l)\Lambda_y(l))V_k \tag{4.35}$$

Having the gradient of objective function we can now calculate the search direction using the steepest descent or conjugate gradient methods, and the update of V can be done through equations (3.16),(3.17) and (3.18). Below is a summary of the algorithm using conjugate gradient over Stiefel manifold.

BSS Algorithm Using Conjugate
Gradient on Stiefel Manifold

Step1 Given the observed data $\mathbf{x}(m)$ form the estimated covariance matrix $\hat{R}_x = \frac{1}{N}\sum_{m=0}^{N-1}\mathbf{x}(m)\mathbf{x}(m)^T$ (N number of samples) and calculate the eigen decomposition:

$$\hat{R}_x = U\Sigma U^T$$

Step2 Calculate the whitening matrix W as:

$$W = \Sigma^{-1}U^T$$

and apply W to observed data to obtain the whitened data $\mathbf{z}(m) = W\mathbf{x}(m)$. Calculate the estimated covariance matrices $R_z(l)$ from

$$\hat{R}_z(l) = \frac{1}{N}\sum_{m=0}^{N-1}\mathbf{z}(m)\mathbf{z}(m-l)^T \qquad l = 0, \cdots, L$$

Step3 Initialize V_0 to some random matrix such that $V_0^T V_0 = I$ and calculate the output $\mathbf{y}(n) = V_0\mathbf{x}(n)$. Calculate the estimated output covariance matrices form

$$\hat{R}_y(l) = V_0\hat{R}_z(l)V_0^T \qquad l = 0, \cdots, L$$

Step4 Calculate:

$$\Gamma_{V_0} = -2\sum_{l=1}^{L}\Lambda_y(l)V_k R_z(l),$$

where $\Lambda_y(k)$ is calculated from (4.34) and from there calculate the gradient:

$$G_0 = \Gamma_{V_0} - V_0\Gamma_{V_0}^T V_0,$$

and set $H_0 = -G_0$

Step5 Choose a value for t $(0 < t < 1)$ and compute the updated value of V from:

$$V_1 = V_0 E(t) + QF(t)$$

where Q and R are the QR decomposition of $(I - V_0V_0^T)H_0$, $A = V_0^T$ and $E(t)$ and $F(t)$ are n-by-n matrices calculated from matrix exponential:

$$\begin{pmatrix} E(t) \\ F(t) \end{pmatrix} = exp\left[t\begin{pmatrix} A & -R^T \\ R & 0 \end{pmatrix}\begin{pmatrix} I_n \\ 0 \end{pmatrix}\right]$$

Step6 Compute $G_1 = \Gamma_{V_1} - V_1\Gamma_{V_1}^T V_1$

Step7 Compute parallel transported tangent vector H_k from point V_0 to point V_1 from:

$$\tau H = H_0E(t) - V_0R^TF(t)$$

and compute the new search direction

$$H_1 = -G_1 + \beta\tau H$$

where

$$\beta = \frac{\langle G - G_0, G\rangle}{\langle G, G\rangle}$$

and $\langle\Delta_1, \Delta_2\rangle = Tr(\Delta_1^T(I - \frac{1}{2}VV^T)\Delta_2)$ represents the inner product between two tangent vectors on Stiefel manifold defined by (3.27).

Step8 Set $V_0 = V_1$, $H_0 = H_1$, $G_0 = G_1$ and repeat steps $5 - 7$. Reset $H_1 = -G_1$ if the number of iterations Mod n(n-1)/2 =0. The iterations can be stopped when $\langle G_0, G0\rangle \leq \epsilon$ where ϵ is a small positive number.

---

5. SIMULATION RESULTS

To measure the performance of the new algorithm and also to compare the speed of convergence between conjugate gradient and gradient descent we applied the algorithm to blind separation of deterministic source signals. For this purpose we used synthetic signals generated by following formulas:
$s_1(m) = sign(\cos(2\pi m/30))$
$s_2(m) = chirp(m, 10, 1000, 1000)$
$s_3(m) = \sin(2\pi m/10 + 6\cos(2\pi m/50))$
$s_4(m) = \sin(2\pi m/10)$
Figure (1) shows the plot of these sources for 100 samples.

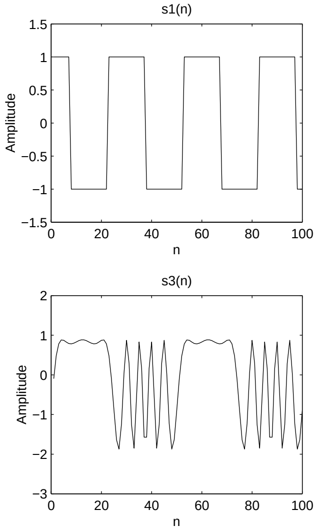

Figure 1: Set of original sources used in the experiment

We mixed the sources by a randomly generated mixing matrix A given by:

$$A = \begin{bmatrix} -0.4977 & -0.7562 & -0.9812 & -0.4129 \\ -1.1187 & -0.0891 & -0.6885 & -0.5062 \\ 0.8076 & -2.0089 & 1.3395 & 1.6197 \\ 0.0412 & 1.0839 & -0.9092 & 0.0809 \end{bmatrix},$$

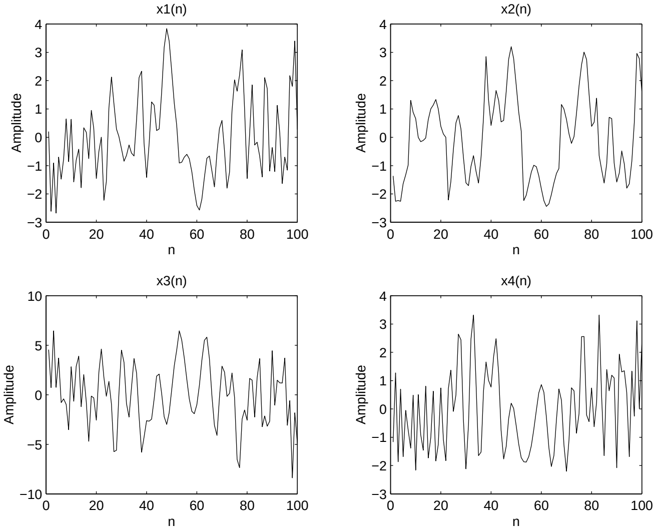

Figure 2: Set of randomly mixed sources (observed signals)

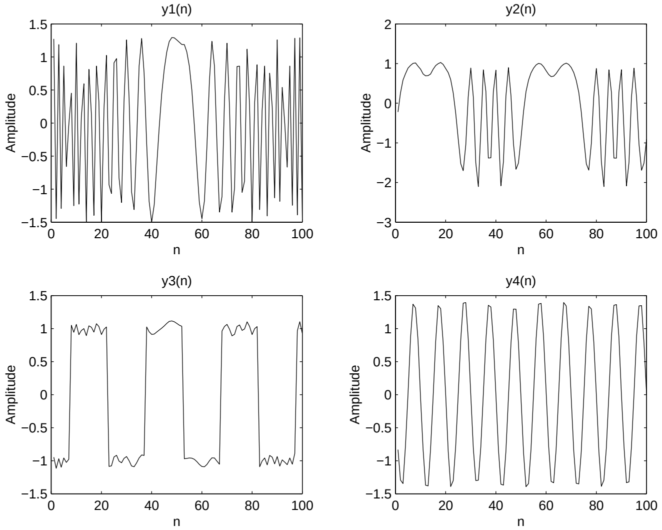

Figure 3: Set of output signals using Polak-Ribiere conjugate gradient method

where the results of mixture, the observed signals $\mathbf{x}(m)$, are shown in figure (2). Separation was done using gradient descent and conjugate gradient on Stiefel manifold. In all experiments V_0 was initialized to an identity matrix. To estimate the covariance matrices in each iteration we used 10000 samples of data. The performance of separation was measured using the formula:

$$P_{index} = 20log_{10}(\frac{1}{n}(\sum_{i=1}^{n}(\sum_{j=1}^{n}\frac{|p_{ij}|}{max_k(|p_{ik}|)} - 1))) \qquad (5.36)$$

where p_{ij} is the (i,j)th element of the matrix $P = VWA$. Notice that for unit power sources, equation (5.36) with good approximation represents the average interference to signal ratio for all the outputs. Figure (4) shows the speed of convergence for gradient descent and the conjugate gradient methods over the Stiefel manifold. The curves show the reduction of P_{index} versus number of iterations. As can be seen from the figure, both conjugate gradient methods converge after 9 iterations.

Figure (3) shows the outputs for the Polak-Ribiere conjugate gradient method. As can be seen, the outputs with good approximation resemble the original sources.

6. CONCLUSIONS

In this paper we applied a geometrical optimization method to blind source separation (BSS) of signals. To do this we first showed that the BSS problem can be modelled as an optimization problem with an orthogonality constraint. The orthogonality constraint represents a nonlinear space known as Stiefel manifold. The idea is that orthonormal constrained optimization problems defined in linear space can be treated as an unconstrained ones on the nonlinear Stiefel manifold. To minimize (or maximize) a function on the Stiefel manifold we can use similar methods for unconstrained optimization in linear space, with the difference that we need to use differential geometry techniques to redefine operations such as gradient, Hessian etc., to appropriate ones on the manifold. Based on these ideas, we developed an adaptive algorithm which uses the steepest descent

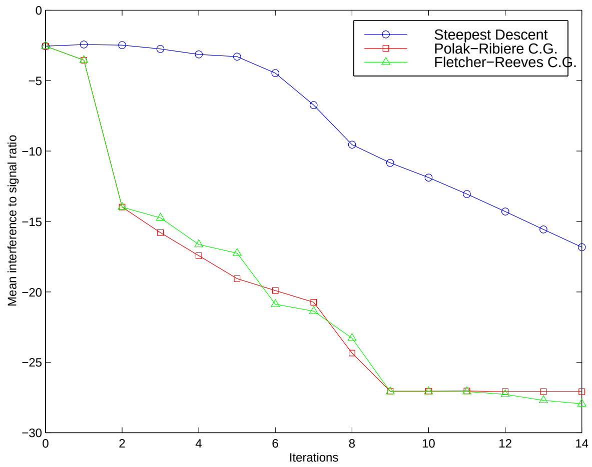

Figure 4: Convergence performance for various forms of optimization over the Stiefel manifold.

and conjugate gradient methods to maximize the objective function. We defined the objective function based on second order moments of the output signals. We applied the new algorithm for blind separation of deterministic sources. The simulation results showed quadratic convergence and good separation performance for the new algorithm.

7. ACKNOWLEDGMENTS

The authors are grateful for financial support from the following institutions: Mitel Corporation, Kanata, Ontario, Canada, the Centre for Information Technology Ontario (CITO), and the Natural Sciences and Engineering Research Council of Canada (NSERC). Helpful input from Prof. Tom Luo, of the Department of Electrical and Computer Engineering, McMaster University, is also acknowledged.

8. REFERENCES

[1] A. Edelman, T. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, Feb. 1998.

[2] P. Comon, “Independent component analysis, A new concept?,” *Signal Processing*, vol. 36, pp. 287–314, 1994.

[3] J. Cardoso and B. Laheld, “Equivariant adaptive source separation,” *Signal Processing*, vol. 44, pp. 3017–3030, Dec. 1996.

[4] J. Cardoso and A. Souloumiac, “Blind beamforming for non Gaussian signals,” in *Proc. IEE-F vol. 140 pp. 362-370*, Dec 1993.

[5] A. Belouchrani, K. Meraim, and J. Cardoso, “A blind source separation technique using second order statistics,” *Signal Processing*, vol. 45, pp. 434–444, Feb. 1997.

[6] Y. Xiang, K. A. Meraim, and Y. Hua, “Adaptive blind source separation by second order statistic and natural gradient,” in *Proc. ICASSP vol. 5 pp. 2917-2920*, (March), Dec 1999.

[7] K. Rahbar and J. P. Reilly, “Blind Source Separation By Minimization of Mutual Information Rate : A second Order Statistics Approach,” in *Submitted to ICASSP*, June 2000.

[8] L. Tong, R. Liu, V. Soon, and Y. Huang, “Indeterminacy and identifiability of blind identification,” *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 499–509, May 1991.

[9] G. Golub and C. VanLoan, *Matrix Computations*. Baltimore and London: John Hopkins, third ed., 1996.

[10] D. Bertsekas, *Nonlinear Programming*. Belmont Mass.: Athena Scientific, second ed., 1999.