A ROBUST AND EFFICIENT ALGORITHM FOR SOURCE PARAMETER ESTIMATION

F. Gersemsky, B. Yang

Department of Electrical Engineering, Ruhr University Bochum, Germany

1. ABSTRACT

A large number of array processing applications such as radar, sonar, etc require the estimation of some parameters given the output of an array of sensors. Many high resolution methods for source parameter estimation are based on the eigen decomposition of the covariance matrix of the sensor output. The PASTd (projection approximation subspace tracking with deflation) algorithm [6] has been recently published for tracking both the signal subspace and its rank at a computational cost of order $O(nr)$, where n is the number of sensors and r the number of sources to be detected. In this paper we address the problem of tracking the physical parameters as direction, distance, etc given the estimated signal subspace. All known parameter estimation methods as MUSIC, MinNorm or WSF are based on a different cost function which is minimized with respect to the desired parameters. Standard minimization methods as gradient or Newton's method fail to converge to the global minimum if the starting value is not close enough to the desired solution [5]. We introduce a new cost function which has to be minimized with respect to the parameters and an algorithm of low computational cost which is able to find the global minimum, starting from any initial value in all our experiments.

2. INTRODUCTION

We present a new cost function which, if minimized with respect to the desired parameters of all sources $\underline{\theta} = [\theta_1, \ldots, \theta_r]$, yields estimates for those parameters. The minimum of this function is found and tracked by an algorithm based on an approximated Newton method combined with a random search for global convergence. Its computational complexity is of order nr^2, where n and r denote the number of sensors and the number of sources, respectively.

Consider an array of n sensors and wavefronts from r narrowband emitters impinging on it. Because of the extension of the array each sensor receives a delayed version of the emitted signal. For narrowband signals this delay appears as a complex phase shift in the baseband signal. The noise associated with each sensor is assumed to be uncorrelated from sensor to sensor. The baseband model of the sensor array can be formulated as the sum of r wavefronts corrupted by white sensor noise

$$\underline{x}(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix} = \sum_{i=1}^{r} s_i(t)\underline{a}(\theta_i) + \underline{n}(t), \quad (1)$$

where $s_i(t)$ and $\underline{a}(\theta_i)$ denote the narrowband signal and the steering vector of complex phase shifts of source i, respectively. The array is assumed to be unambiguous, meaning that the steering vectors of n sources with different parameters θ_i $(i = 1, \ldots, n)$ are linearly independent. The derivative of a steering vector with respect to θ is denoted as $\frac{\partial \underline{a}}{\partial \theta} = \underline{d}$. The white noise is represented as $\underline{n}(t)$. If the white noise assumption does not hold, the noise covariance matrix $\mathrm{E}\underline{n}\,\underline{n}^H = C_N$ can be estimated and the signals can be prewhitened with $\tilde{\underline{x}}(t) = \hat{C}_N^{-1/2}\underline{x}(t)$.

Here we restrict ourselves to the case of one parameter per source. Extension to multiple parameters is straightforward but leads to a more complicated formulation of the algorithm. The covariance matrix of $\underline{x}(t)$ reads

$$C_X = \mathrm{E}\underline{x}\,\underline{x}^H = A(\underline{\theta})C_S A^H(\underline{\theta}) + \sigma^2 I \quad (2)$$

with $A(\underline{\theta}) = [\underline{a}(\theta_1), \ldots, \underline{a}(\theta_r)]$ and C_S being the matrix of steering vectors and the source covariance matrix, respectively. We assume the signals to be not fully correlated leading to a full rank matrix C_S. The noise power is denoted as σ^2.

The eigen decomposition of C_X is given as

$$C_X = \sum_{i=1}^{n} \lambda_i \underline{u}_i\,\underline{u}_i^H = U_S \Lambda_S U_S^H + U_N \Lambda_N U_N^H. \quad (3)$$

The matrix U_S contains the r most dominant eigenvectors which span the so called signal subspace. It can be easily verified that the subspaces spanned by the

0-7803-2431-5/95 $4.00

columns of U_S and $A(\underline{\theta})$ are identical. This property has been used in most of the high resolution methods and will also be used here.

3. THE NEW COST FUNCTION

The basic idea is to find a set of parameters $\underline{\theta}$ such that the subspaces spanned by U_S and $A(\underline{\theta})$ are as close as possible. To accomplish for this we define the following cost function which has to be minimized with respect to $\underline{\theta}$

$$h(\underline{\theta}) = \sum_{i=1}^{r} f(\theta_i) g(\underline{\theta}), \tag{4}$$

where

$$f(\theta_i) = \underline{a}^H(\theta_i) P_S^{\perp} \underline{a}(\theta_i) \quad \text{and} \tag{5}$$

$$g(\underline{\theta}) = \left\| (A^H(\underline{\theta}) A(\underline{\theta}))^{-1} \right\|_2 . \tag{6}$$

$P_S^{\perp} = I - U_S (U_S^H U_S)^{-1} U_S^H$ denotes the projection matrix onto the null space of U_S . The function $f(\theta_i)$ is a measurement of the distance between the steering vector $\underline{a}(\theta_i)$ and the signal subspace U_S. If $\underline{a}(\theta_i)$ lies in the signal subspace, $f(\theta_i)$ will be zero indicating that θ_i matches one of the source parameters.

The matrix 2-norm [2] in $g(\underline{\theta})$ increases whenever the matrix $A^H A$ is close to singular. Minimizing $h(\underline{\theta})$ therefore forces A to have full rank. Another reason for this choice of $g(\underline{\theta})$ is that this function is approximately constant whenever all elements of $\underline{\theta}$ are distinct. This ensures that the bias introduced by the function g is small compared to the stochastic error introduced by the noisy estimates of the signal subspace. If we chose $g(\underline{\theta}) \equiv 1$, then $1/h(\underline{\theta})$ would be a multidimensional MUSIC-function [3]. Figure 1 shows a 3-D plot of $1/h(\underline{\theta})$ with $g(\underline{\theta}) \equiv 1$ for the case where two sources are located at $\theta_1 = 35°$ and $\theta_2 = -20°$. This function exhibits four dominant maxima at positions $\underline{\theta}$ = [35° -20°], [-20° 35°], [35° 35°], [-20° -20°] leading to the well known problem of deciding, which solution to choose. In this case, only the first two possible solutions yield satisfying results. The latter two lead to a solution where one parameter is detected twice and the other one is lost. In contrast, Figure 2 depicts $1/h(\underline{\theta})$ for the case where $g(\underline{\theta})$ has been chosen according to Eq. (6). It can be clearly seen that the two undesirable peaks have vanished. Locating one of the two remaining peaks always leads to the desired parameters.

The computation of the 2-norm of a matrix requires the calculation of the most dominant singular value of this matrix, a task which is iterative and computationally expensive. Additionally, the derivatives of

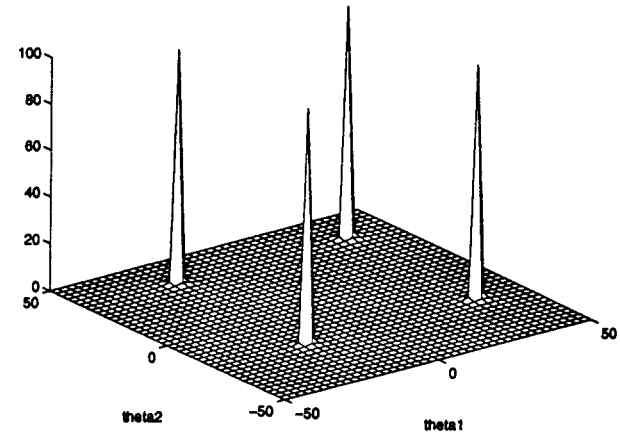

Figure 1: Cost function of MUSIC

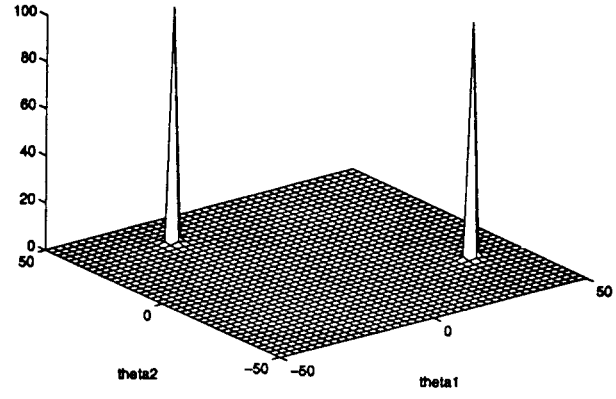

Figure 2: New cost function

the 2-norm with respect to $\underline{\theta}$ are in general not available. Hence we use a different approach which proved its usefulness in various simulations. Because $A^H A$ is hermitian and positive definite, its least dominant singular vector and the most dominant eigenvector of $(A^H A)^{-1}$ are identical. This most dominant eigenvector denoted by $\underline{v}$ can be computed iteratively using the well known power method [2]. Choosing an arbitrary initial value for $\underline{v}_0$, the following update formula will converge to the most dominant eigenvector of the matrix $(A^H A)^{-1}$

$$\tilde{\underline{v}}_k = (A^H A)^{-1} \underline{v}_{k-1} \tag{7}$$

$$\underline{v}_k = \frac{\tilde{\underline{v}}_k}{\|\tilde{\underline{v}}_k\|}. \tag{8}$$

Various simulations showed that it is sufficient to perform only one iteration each sample and we still get a good approximation for the eigenvector and eigenvalue. If we assume that $\underline{v}$ is a good estimate for the

most dominant eigenvector of $(\boldsymbol{A}^H\boldsymbol{A})^{-1}$ with the corresponding eigenvalue λ_1, $g(\underline{\theta})$ can now be written as

$$g(\underline{\theta}) = \lambda_1 = \underline{v}^H(\boldsymbol{A}^H\boldsymbol{A})^{-1}\underline{v} = \left(\underline{v}^H\boldsymbol{A}^H\boldsymbol{A}\underline{v}\right)^{-1}. \quad (9)$$

4. THE APPROXIMATED NEWTON METHOD

For updating the parameter estimates, we use a Newton type algorithm. The update formula for the parameter vector $\underline{\theta}$ can be written as

$$\underline{\theta}(t) = \underline{\theta}(t-1) - \boldsymbol{H}_h^{-1}\underline{\nabla}h\big|_{\underline{\theta}=\underline{\theta}(t-1)}, \quad (10)$$

where $\underline{\nabla}h(\underline{\theta})$ and $\boldsymbol{H}_h(\underline{\theta})$ are the gradient and the Hessian matrix of $h(\underline{\theta})$. The gradient of $h(\underline{\theta})$ evaluates to

$$\underline{\nabla}h(\underline{\theta}) = \underline{\nabla}g(\underline{\theta})f(\underline{\theta}) + \begin{pmatrix} \frac{\partial f(\theta_1)}{\partial\theta_1} \\ \vdots \\ \frac{\partial f(\theta_r)}{\partial\theta_r} \end{pmatrix} g(\underline{\theta}) \quad (11)$$

with $\frac{\partial f(\theta)}{\partial\theta} = 2\Re[\underline{d}^H(\theta)\boldsymbol{P}_S^{\perp}\underline{a}(\theta)]$.

To compute the gradient of $g(\underline{\theta})$ we have to consider the following. With $\underline{v}$ being the most dominant eigenvector of some positive definite and hermitian matrix $\boldsymbol{B}$, the 2-norm of $\boldsymbol{B}$ can be expressed as $\|\boldsymbol{B}\|_2 = \lambda_1 = \underline{v}^H\boldsymbol{B}\underline{v}$. The derivative of λ_1 with respect to some parameter θ reads

$$\begin{aligned} \frac{\partial\lambda_1}{\partial\theta} &= \frac{\partial\underline{v}^H}{\partial\theta}\boldsymbol{B}\underline{v} + \underline{v}^H\frac{\partial\boldsymbol{B}}{\partial\theta}\underline{v} + \underline{v}^H\boldsymbol{B}\frac{\partial\underline{v}}{\partial\theta} \quad (12)\\ &= \lambda_1\left(\frac{\partial\underline{v}^H}{\partial\theta}\underline{v} + \underline{v}^H\frac{\partial\underline{v}}{\partial\theta}\right) + \underline{v}^H\frac{\partial\boldsymbol{B}}{\partial\theta}\underline{v} \quad (13)\\ &= \underline{v}^H\frac{\partial\boldsymbol{B}}{\partial\theta}\underline{v}, \quad (14)\end{aligned}$$

because of $\frac{\partial(\underline{v}^H\underline{v})}{\partial\theta} = 0$ for a normalized eigenvector $\underline{v}$. Using this result, we can compute $\frac{\partial g(\underline{\theta})}{\partial\theta_i}$

$$\begin{aligned} \frac{\partial g(\underline{\theta})}{\partial\theta_i} &= \underline{v}^H\frac{\partial(\boldsymbol{A}^H\boldsymbol{A})^{-1}}{\partial\theta_i}\underline{v}\\ &= -\underline{v}^H(\boldsymbol{A}^H\boldsymbol{A})^{-1}\frac{\partial(\boldsymbol{A}^H\boldsymbol{A})^{-1}}{\partial\theta_i}(\boldsymbol{A}^H\boldsymbol{A})^{-1}\underline{v}\\ &= -\lambda_1^2\underline{v}^H\frac{\partial(\boldsymbol{A}^H\boldsymbol{A})^{-1}}{\partial\theta_i}\underline{v} \quad (15)\end{aligned}$$

and

$$\underline{\nabla}g(\underline{\theta}) = -2\lambda_1^2\Re[(\boldsymbol{D}^H\boldsymbol{A}\underline{v})\odot(\underline{v}^*)], \quad (16)$$

where $\odot$ denotes elementwise multiplication and $\boldsymbol{D}(\underline{\theta})$ is defined as $\boldsymbol{D}(\underline{\theta}) = [\underline{d}(\theta_1),\ldots,\underline{d}(\theta_r)]$.

Instead of using the exact expression for the Hessian matrix, we use the approximation $\tilde{\boldsymbol{H}}_h = g\tilde{\boldsymbol{H}}_f$ where $\tilde{\boldsymbol{H}}_f$ is a diagonal matrix with entries $\tilde{\boldsymbol{H}}_f\{i,i\} = 2\,\underline{d}^H(\theta_i)\boldsymbol{P}_S^{\perp}\underline{d}(\theta_i)$. This approximation offers several advantages:

- The approximation is close to the exact value in the neighbourhood of the desired parameter.
- The matrix $\tilde{\boldsymbol{H}}_h$ is positive definite.
- There is no need to compute the second derivatives of $g(\underline{\theta})$ which would require the full eigen decomposition of $\boldsymbol{A}^H\boldsymbol{A}$.
- Because of $\tilde{\boldsymbol{H}}_h$ being a diagonal matrix, its inversion reduces to r scalar divisions.

The computational complexity for the update of the parameter vector $\underline{\theta}$ is hence of the order $O(nr^2)$.

If we simply used this Newton type algorithm to iteratively search for the minimum of $h(\underline{\theta})$, we would have the well known problem of local convergence. This would require good initial guesses for the parameters which are in general not available. To avoid this problem we introduce a combination of Newton's method with a random search based on some properties of the function f. It is easy to show that the value of $f(\theta_i)$ is almost 0 if θ_i matches the desired parameter of one of the sources. In any other cases, its value is approximately n. This leads to the idea of performing a random jump for those parameters θ_i whose values $f(\theta_i)$ are greater than some bound f_{Max}. In our examples we choose $f_{Max} = 0.05n$.

In order to not disturb the Newton method from converging to the desired solution by random jumps, those jumps are only allowed if the absolute value of the derivative of $f(\theta_i)$ is below a certain bound f'_{Max} indicating that the algorithm is close to a local minimum. This bound has to be chosen according to f_{Max}. If $|f'(\theta_i)|$ is below f'_{Max} and the algorithm is going to converge to a valid solution, then the value of $f(\theta_i)$ has to be smaller than the bound f_{Max} to prevent random jumps in this case. In our simulations, a value of $f'_{Max} = 0.1$ lead to reasonable results.

Now we can formulate our algorithm

FOR $i = 1$ TO r DO

$$\theta_i(t) = \theta_i(t-1) - \frac{g(\underline{\theta})\frac{\partial f(\theta_i)}{\partial\theta_i} + f(\underline{\theta})\frac{\partial g(\underline{\theta})}{\partial\theta_i}}{\tilde{\boldsymbol{H}}_h\{i,i\}}$$

IF $(f(\theta_i) > f_{Max})$ AND $(|f'(\theta_i)| < f'_{Max})$

THEN $\theta_i = X_k$

END IF

END FOR

X_k are random variables uniformly distributed over the parameter range of θ.

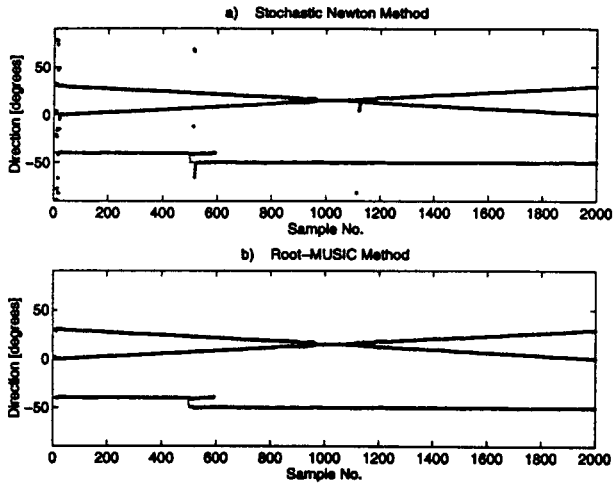

Figure 3: Simulation Results

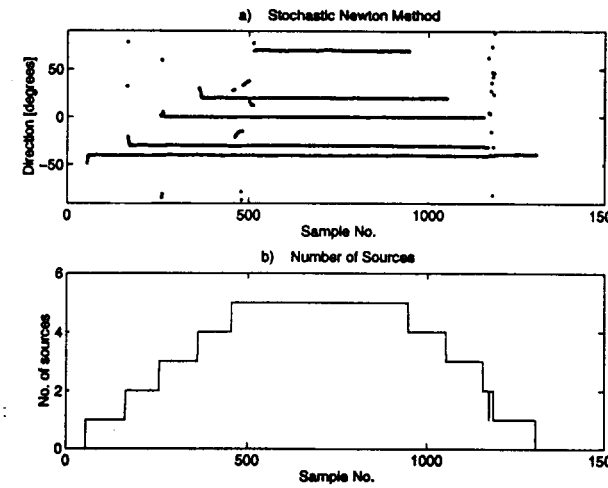

Figure 4: Simulation Results

5. RESULTS

A linear equispaced array consisting of $n = 9$ sensors has been used in all simulations. We used the PASTd algorithm [6] for the signal subspace and rank estimation. Our method is compared with a Root-MUSIC approach [1].

The first scenario shows three sources – two of them crossing at t = 1000 and the third one changing its position instantaneously at t = 500. The signal to noise ratio (SNR) is 0 dB. The simulation shows that our method (Figure 3a) performs as well as the Root-MUSIC method (Figure 3b) whose computational complexity is of order $O(n^3)$. The new method only requires a few samples to identify the locations of the sources and each time the PASTd algorithm decides to change the number of sources, our method drops the right one or finds the new source at the right position. The second example consists of five sources which appear and disappear one after the other. Figure 4a shows the estimated directions and Figure 4b depicts the number of sources that the PASTd algorithm detects. It can be clearly seen that our algorithm detects each newly appearing source at most 50 samples after the PASTd algorithm realized it.

6. CONCLUSION

The results show that our new method is able to detect sources with the same performance as the computationally much more expensive Root-MUSIC method. Even more important is the fact that this method is not restricted to linear arrays as Root-MUSIC is. In contrast to other parameter estimation methods, it is not only local convergent. In all simulations the algorithm converged to the desired solution from any starting point. The same algorithm can be applied to the frequency estimation problem of time series. The analogy between these two problems has been extensively studied in [3].

7. REFERENCES

[1] A. J. Barabell, "Improving the Resolution Performance of Eigenstructure-based Direction Finding Algorithms", Proc. IEEE ICASSP, pp. 336-339, Boston, April 1983.

[2] G. H. Golub, "Matrix Computations", The Johns Hopkins University Press, Baltimore, 1990

[3] S. Haykin, "Adaptive Filter Theory", Prentice Hall, 1991

[4] T. Kato, "Perturbation Theory for Linear Operators", Springer-Verlag, Berlin, 1980

[5] Mats Viberg, "Sensor Array Processing Based on Subspace Fitting", IEEE Transactions on Signal Processing, Vol 39, No 5, May 1991

[6] Bin Yang, "An Adaptive Algorithm of Linear Computational Complexity for both Rank and Subspace Tracking", Proc. IEEE ICASSP, pp. IV33-IV36, Adelaide, Apr. 1994.