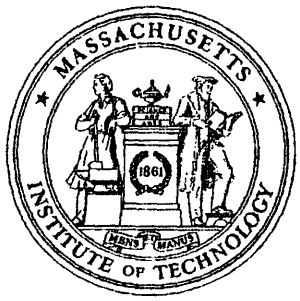


Iterative Maximum Likelihood Time Delay and Doppler Estimation Using Stationary Signals

RLE Technical Report No. 545

September 1989

Bruce R. Musicus and Ehud Weinstein

Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Iterative Maximum Likelihood Time Delay and Doppler Estimation Using Stationary Signals[3]

Bruce R. Musicus[1] Ehud Weinstein[2]

September 4, 1989

Abstract

We develop computationally efficient iterative algorithms for the joint Maximum Likelihood (ML) estimation of the time delays, Doppler shifts, and spectral parameters of stationary Gaussian signals radiated from a stationary or moving point source and observed in the presence of uncorrelated additive noise at two or more spatially distributed receivers. Perhaps the most striking feature of these algorithms is that they decompose the estimation of the signal spectral parameters from the estimation of the delay and Doppler parameters, leading to a considerable simplification in estimator structure and computation. The proposed algorithms converge to the set of stationary points of the likelihood function, and each iteration increases the likelihood. All algorithms are derived from a common iterative framework related to the Estimate-Maximize algorithm, and we analyze their convergence rates both theoretically and via simulation.

Index Terms: Time-delay estimation, Doppler Shift Estimation, Maximum Likelihood, Estimate-Maximize (EM) Algorithm, Iterative Algorithms.

---

[1]Room: 36-789, 50 Vassar Street, Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, Tel: (617) 253-8845.

[2]Department of Electronic Systems, Faculty of Engineering, Tel-Aviv University, Ramat Aviv, Tel-Aviv 69978, ISRAEL.

[3]This work has been supported by the USA AMCCOM, ARDEC, U.S. Army Research Office managed by Battelle under Prime Contract No. DAAL03-86-D-0001, and in part by the Advanced Research Project Agency monitored by ONR under Contract No. N00014-85-K-0272.

Contents

1 Introduction

Time delays between signals radiated from a common point source and observed at two or more spatially separated receivers can be used to determine source location. Time delay estimation has therefore attracted a great deal of interest in the literature (e.g. [1]). Most of the analyses assume that the source signal and the additive receiver noises are mutually independent wide sense stationary (WSS) Gaussian processes with known spectra, and that the observation interval is long compared with the correlation time (inverse bandwidth) of the signal and the noises. In that case, the Maximum Likelihood (ML) estimate of the receiver-to-receiver delay is obtained by pre-filtering and cross-correlating the received signals, and searching for the peak of the cross-correlator response [13,10,7,12]. Under the stated assumptions, the ML delay estimate is optimal in the sense that it is asymptotically unbiased and its error variance approaches the Cramer-Rao lower bound.

If there are $M > 2$ receivers, there are $(M-1)$ linearly independent differential delays to be estimated. Joint ML estimation of the these delays requires a search over an $(M-1)$-dimensional space of delay values. An alternative approach that avoids this multi-dimensional optimization consists of independently estimating the $M(M-1)/2$ differential delays between all receiver pairs, and then using a linear least squares fit to convert these estimates into estimates of the $(M-1)$ linearly independent delays [8]. However, this approach requires $M(M-1)/2$ cross-correlators, which may be prohibitive for large arrays.

If the source is moving relative to the array, the signals observed at different receivers are not only time delayed but also time compressed relative to each other. Measurement of these Doppler time compression coefficients provides important additional information concerning source location, velocity, and heading. However, the time scaling effect causes the signals observed at different receivers to be jointly non-stationary, and thus complicates the estimation problem quite drastically. An approximate ML scheme was developed in [11]. Basically, it forms the cross-correlation of one receiver output with respect to a time-delayed and time-scaled version of the other receiver output, and

obtains the joint ML estimate of the delay and Doppler parameters by maximizing the cross-correlation response. In the multiple receiver case with $M > 2$, we need to estimate all $(M-1)$ pairs of differential delay and Doppler parameters jointly, and the amount of computation required increases substantially.

In all of these analyses, it is assumed that the signal and noise spectra are known *a priori*. In practice, this is apt to be unrealistic. One is unlikely to have accurate prior information about signal bandwidth, center frequency, or power level, and the spectral description of the noise field may be similarly incomplete. It has been shown [19] that lack of knowledge of spectral parameters does not degrade the quality (mean square error) of the delay estimate, provided that the joint ML estimation of the delay and the spectral parameters is carried out. Unfortunately, for most cases, the joint ML estimation involves a complicated multi-dimensional optimization that is difficult to solve. A common sub-optimal approach consists of estimating the signal and noise spectra (or alternatively, the coherence function), and using these to construct the pre-filters to be used prior to the cross-correlation operation (e.g. [7,12,3,4,9]). However, this procedure is ad-hoc, and its inherent accuracy critically depends on the method employed for spectral estimation.

In this paper we develop computationally efficient schemes for joint ML estimation of the delays, Dopplers, and spectral parameters, all based on different variants of the iterative Estimate-Maximize (EM) algorithm. Perhaps the most striking feature of these proposed algorithms is that they decompose the estimation of the spectral parameters from the estimation of the delay parameters without any sacrifice in estimation accuracy (mean square error). In the multiple receiver case, we develop an algorithm that further simplifies the problem, replacing the multidimensional optimization over the vector of delays and Dopplers with an optimization which estimates each pair of delay and Doppler parameters independently. The proposed algorithms increase the likelihood on every iteration, and they converge to the set of stationary points and local maxima of the likelihood function. Their convergence rates can be analyzed theoretically.

The organization of this report is as follows. In section 2 we present the ML problem

of estimating the delays and spectral parameters of a stationary (non-moving) source observed by an array of spatially separated receivers. We show that the direct ML solution is quite difficult to compute. We briefly discuss some existing sub-optimal solutions and ad-hoc approaches to the problem. In section 3, we develop iterative algorithms for solving ML problems based on an approach related to the Estimate-Maximize (EM) algorithm. The proposed algorithms are optimal in the sense that they converge iteratively to a stationary point of the likelihood function, and each iteration increases the likelihood of the estimated parameters. In section 4 we apply the simplest algorithm to the time delay estimation problem. In section 5 we analyze the convergence behavior of this algorithm for the case of very long observation intervals, and show that the delay and spectral estimates converge linearly to the desired ML solution. In section 6, we present several hybrid EM-ML iterative algorithms, which require more computation, but which achieve super-linear convergence rates. Section 7 contains simulation results for the algorithms, and compares their performance (mean square error) with the Cramer-Rao lower bound. In section 8 we consider the problem of estimating the additional Doppler parameters caused by relative motion between source and receivers. We confine our attention to stationary narrowband signals, in which case the Doppler effect effectively causes a frequency shift of the signals observed at the various receiver outputs. We first present the direct ML approach to the problem, followed by computationally efficient EM and EM-ML hybrid algorithms. Finally, in section 9 we summarize the results.

2 Problem Formulation and Existing Results

Signals radiated from a stationary (non-moving) point source, propagating through a non-dispersive medium, and observed in the presence of additive noise by M spatially distributed receivers, can be modeled by:

$$z_i(t) = \alpha_i s(t - \tau_i) + v_i(t) \qquad \begin{array}{l} T_I \leq t \leq T_F \\ i = 1, \ldots, M \end{array} \tag{1}$$

where τ_i is the travel time of the signal $s(t)$ from the source to the i^{th} receiver, and α_i is the amplitude attenuation of the signal wavefront at the i^{th} receiver. In many applications, $\alpha_i \geq 0$.

We suppose that $s(t)$ and $v_i(t)$ for $i = 1, \ldots, M$ are mutually independent, jointly wide-sense stationary Gaussian random processes with power spectra $P_S(\omega; \underline{\theta})$ and $P_{V_i}(\omega; \underline{\sigma}_i)$ respectively. The vectors $\underline{\theta}$ and $\underline{\sigma}_i$ for $i = 1, \ldots, M$ represent possibly unknown signal and noise spectral parameters such as bandwidth, center frequency, fundamental frequency, average power level, pole/zero coefficients, etc.

Given continuous or discrete-time observations of the receiver outputs $z_1(t), \ldots, z_M(t)$ for $T_I \leq t \leq T_F$, we want to find the Maximum Likelihood (ML) estimate of the delays τ_i. Because the likelihood function also depends on the unknown gains and spectral parameters, we will have to estimate these also. Let $\underline{\tau}$, $\underline{\alpha}$, $\underline{\sigma}$ represent the vectors of unknown delays, signal gains, and noise spectral parameters, respectively, and let $\underline{\xi}$ represent the vector of all unknown parameters:

$$\underline{\xi} = \begin{pmatrix} \underline{\tau} \\ \underline{\alpha} \\ \underline{\sigma} \\ \underline{\theta} \end{pmatrix} \tag{2}$$

Because $s(t)$ is unknown and stationary, we can only identify the relative delays $\tau_i - \tau_j$. In the following we will sometimes explicitly recognize this by setting one of the delays to a fixed value, e.g. $\tau_M = 0$. If the model for $P_S(\omega; \underline{\theta})$ allows adjusting the overall signal spectral level, we may also wish to fix at least one of the gains to a fixed value,

e.g. $\alpha_M = 1$, or constrain the energy in the gains, $\sum_i \alpha_i^2 = \kappa$ for some constant κ. Because the vectors $\underline{\tau}$, $\underline{\alpha}$, $\underline{\sigma}$, and $\underline{\theta}$ will only contain the unknown parameters, their dimensions may vary depending on how much is known about the application.

Note that in most beam forming applications with $M > 2$ receivers, the fixed geometry of the array constrains the feasible set of delays τ_i. Rather than taking arbitrary values, these delays may be restricted to being deterministic functions, $\tau_i = \tau_i(\rho, \phi)$, of the source range ρ and bearing ϕ. In this case, the target location estimates may be found by directly solving for the ML estimates of bearing and range. This results in a lower dimension search for the optimum, and typically gives more robust estimates. Although the methods in this paper can be adapted to estimate bearing and range directly, we will not consider this in the development below.

2.1 Likelihood Function

For stationary signal and noises, it is convenient to state the estimation problem in the frequency domain. Fourier analyzing the various $z_i(t)$:

$$Z_i(\omega_n) = \frac{1}{\sqrt{T}} \int_{T_I}^{T_F} z_i(t) e^{-j\omega_n t}\, dt \tag{3}$$

where $T = T_F - T_I$ and where $\omega_n = 2\pi n/T$ is the n^{th} frequency sample. (In the case of discrete observations, we replace the integral by the appropriate sum.) Similarly, define $S(\omega_n)$ and $V_i(\omega_n)$ as the Fourier transforms of $s(t)$ and $v_i(t)$ respectively. Define $\underline{Z}(\omega_n)$ as the $M \times 1$ vectors of the M receiver coefficients associated with frequency ω_n:

$$\underline{Z}(\omega_n) = \begin{pmatrix} Z_1(\omega_n) \\ \vdots \\ Z_M(\omega_n) \end{pmatrix} \tag{4}$$

Define $\underline{V}(\omega_n)$ similarly.

Fourier transforming the model (1) gives a frequency domain version of the model:

$$Z_i(\omega_n) = \alpha_i e^{-j\omega_n \tau_i} S(\omega_n) + V_i(\omega_n) \tag{5}$$

Using the notation defined above, we can write (5) in a more convenient vector form as follows:

$$\underline{Z}(\omega_n) = \underline{U}\,(\omega_n; \underline{\tau}, \underline{\alpha})\, S(\omega_n) + \underline{V}(\omega_n) \tag{6}$$

where:

$$\underline{U}(\omega_n;\underline{\tau},\underline{\alpha}) = \begin{pmatrix} \alpha_1 e^{-j\omega_n\tau_1} \\ \vdots \\ \alpha_M e^{-j\omega_n\tau_M} \end{pmatrix} \tag{7}$$

Assume that the signal and noises are bandlimited, with maximum bandwidth W. Let us define the data vector $\underline{z}$ as the concatenation of all N frequency samples $\underline{Z}(\omega_n)$ in the band of interest, which for convenience we will assume to be between ω_L and ω_U:

$$\underline{z} = \begin{pmatrix} \underline{Z}(\omega_L) \\ \vdots \\ \underline{Z}(\omega_U) \end{pmatrix} \tag{8}$$

Let $N = U - L + 1$ be the total number of independent frequency samples in the band of interest. We note that since the $z_i(t)$ are real-valued functions, then $Z_i(-\omega_n) = Z_i^*(\omega_n)$ (where $*$ denotes the complex conjugation) so we only need to consider positive frequencies. To simplify the development, we will also ignore the DC term. (Since $\underline{Z}(0)$ is real while all the other frequency samples are complex, the DC term would require special handling.) We will indicate later how to modify the algorithms to include the DC term.

We assume that the observation interval T is long compared with the correlation time (inverse bandwidth $W/2\pi$) of the signal and the noises (i.e. $WT/2\pi \gg 1$). In that case, the vector Fourier coefficients $\underline{Z}(\omega_n)$ associated with different frequencies are statistically independent, multi-variate complex Gaussian random variables with probability densities:

$$p\left(\underline{Z}(\omega_n)\right) = \frac{1}{\det\left[\pi \mathbf{P}_Z(\omega_n;\underline{\xi})\right]} e^{-\underline{Z}^*(\omega_n)\mathbf{P}_Z^{-1}(\omega_n;\underline{\xi})\underline{Z}(\omega_n)} \tag{9}$$

where $\mathbf{P}_Z(\omega_n;\underline{\xi})$ is the data covariance matrix at frequency ω_n:

$$\begin{aligned} \mathbf{P}_Z(\omega_n;\underline{\xi}) &= \mathrm{E}_{\underline{\xi}}\left[\underline{Z}(\omega_n)\underline{Z}^*(\omega_n)\right] \\ &= \underline{U}(\omega_n;\underline{\tau},\underline{\alpha}) P_S(\omega_n;\underline{\theta})\underline{U}^*(\omega_n;\underline{\tau},\underline{\alpha}) + \mathbf{P}_V(\omega_n;\underline{\sigma}) \end{aligned} \tag{10}$$

where $\underline{Z}^*$ is the complex conjugate transpose (Hermitian) of $\underline{Z}$, $\underline{U}(\omega_n;\underline{\tau},\underline{\alpha})$ is defined

in (7), and where $\mathbf{P}_V(\omega_n;\underline{\sigma})$ is an $M \times M$ diagonal matrix of noise covariances:

$$\mathbf{P}_V(\omega_n;\underline{\sigma}) = \begin{pmatrix} P_{V_1}(\omega_n;\underline{\sigma}_1) & & 0 \\ & \ddots & \\ 0 & & P_{V_M}(\omega_n;\underline{\sigma}_M) \end{pmatrix} \tag{11}$$

Invoking the statistical independence of the $\underline{Z}(\omega_n)$, $n = 1,\ldots,N$, the observed log likelihood function is:

$$\begin{aligned} L_Z(\underline{\xi}) &= \log p\left(\underline{Z};\underline{\xi}\right) \\ &\approx \sum_{n=L}^{U} \log p\left(\underline{Z}(\omega_n)\right) \\ &= -\sum_{n=L}^{U} \left[\log\det\left[\pi\mathbf{P}_Z(\omega_n;\underline{\xi})\right] + \underline{Z}^*(\omega_n)\mathbf{P}_Z^{-1}(\omega_n;\underline{\xi})\underline{Z}(\omega_n)\right] \end{aligned} \tag{12}$$

(Note that this formula would be exactly correct if all the processes were periodic with period T.) Substituting (10) into (12) and carrying out the indicated matrix manipulations:

$$\begin{aligned} L_Z(\underline{\xi}) = -\sum_{n=L}^{U} \Bigg[&\sum_{i=1}^{M} \log \pi P_{V_i}(\omega_n;\underline{\sigma}_i) + \log\left(1 + \sum_{i=1}^{M} \frac{|\alpha_i|^2 P_S(\omega_n;\underline{\theta})}{P_{V_i}(\omega_n;\underline{\sigma}_i)}\right) \\ &+ \sum_{i=1}^{M} \frac{|Z_i(\omega_n)|^2}{P_{V_i}(\omega_n;\underline{\sigma}_i)} - \frac{P_S(\omega_n;\underline{\theta})\left|\sum_{i=1}^{M} \alpha_i e^{-j\omega_n\tau_i} Z_i^*(\omega_n)/P_{V_i}(\omega_n;\underline{\sigma}_i)\right|^2}{1 + \sum_{i=1}^{M} \alpha_i^2 P_S(\omega_n;\underline{\theta})/P_{V_i}(\omega_n;\underline{\sigma}_i)} \Bigg] \end{aligned} \tag{13}$$

Computing the ML estimate of the unknown parameters requires maximizing $L_Z(\underline{\xi})$ with respect to all the unknowns $\underline{\xi}$:

$$\hat{\underline{\xi}}_{ML} \leftarrow \max_{\underline{\xi}} L_Z(\underline{\xi}) \tag{14}$$

The ML method is known to be asymptotically efficient. Thus, for $WT/2\pi \gg 1$ and sufficiently high signal-to-noise ratios, $\hat{\underline{\xi}}_{ML}$ is asymptotically unbiased, and its error variance approaches the Cramer-Rao lower bound (CRLB), that is:

$$Cov\left(\hat{\underline{\xi}}_{ML}\right) \approx J^{-1}(\underline{\xi}) \tag{15}$$

where $J(\xi)$ is the Fisher Information Matrix (FIM) defined by:

$$J(\xi) = -\mathrm{E}\left[\frac{\partial^2}{\partial\underline{\xi}^2} L_Z(\underline{\xi})\right] \tag{16}$$

In [19] it is shown that:

$$J(\underline{\xi}) = \begin{bmatrix} J(\underline{\tau}) & 0 \\ 0 & J(\underline{\alpha}, \underline{\sigma}, \underline{\theta}) \end{bmatrix} \tag{17}$$

where $J(\underline{\tau})$ is the FIM associated with the delay parameters:

$$J_{ik}(\underline{\tau}) = \begin{cases} -2 \sum_{n=L}^{U} \frac{\omega_n^2 SNR_i(\omega_n) SNR_k(\omega_n)}{1 + \sum_{m=1}^{M} SNR_m(\omega_n)} & i \neq k \\ 2 \sum_{n=L}^{U} \frac{\omega_n^2 SNR_i(\omega_n) \sum_{m \neq i} SNR_m(\omega_n)}{1 + \sum_{m=1}^{M} SNR_m(\omega_n)} & i = k \end{cases} \tag{18}$$

where $SNR_i(\omega_n)$ is the signal-to-noise spectral ratio at the i^{th} receiver output:

$$SNR_i(\omega_n) = \frac{\alpha_i^2 P_S(\omega_n; \underline{\theta})}{P_{V_i}(\omega_n; \underline{\sigma_i})} \tag{19}$$

The block diagonal form of $J(\underline{\xi})$ has several important implications. First, it asserts that $J^{-1}(\underline{\xi})$ is also block diagonal, indicating that the errors in the $\underline{\tau}$ estimate are asymptotically statistically uncorrelated with the errors in the $\underline{\alpha}$, $\underline{\sigma}$, and $\underline{\theta}$ estimates. It further asserts that:

$$Cov(\hat{\underline{\tau}}) \approx J^{-1}(\underline{\tau}) \tag{20}$$

But $J^{-1}(\underline{\tau})$ is the Cramer-Rao lower bound on the error covariance of $\underline{\tau}$ when $\underline{\alpha}$, $\underline{\sigma}$, and $\underline{\theta}$ are known a-priori. Therefore, if we carry out the *joint* ML estimation of *all* the unknown parameters, then the quality of the delay estimates is not degraded by the errors in the gain and spectral estimates. Unfortunately, for most cases of interest, the joint ML estimation of the delay, gain, and spectral parameters required in (14) involves a complicated multi-parameter optimization that is very difficult to solve.

2.2 Approximate Likelihood Maximization Methods

Because the ML estimation problem is so difficult to solve directly, it is useful to consider approximate methods first. The problem would be simplified, for example, if the signal gains and spectral parameters were known exactly, so that we only needed to optimize with respect to the delay parameters τ_i. In this case, we could write the log likelihood as:

$$L_Z(\underline{\tau}) = c' + 2 \text{ Re} \sum_{n=L}^{U} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} W_{ik}(\omega_n) Z_i(\omega_n) Z_k^*(\omega_n) e^{j\omega_n(\tau_i - \tau_k)} \tag{21}$$

where c' is independent of $\underline{\tau}$, and:

$$W_{ik}(\omega_n) = \frac{\alpha_i \alpha_k P_S(\omega_n;\underline{\theta}) / P_{V_i}(\omega_n;\underline{\sigma_i}) P_{V_k}(\omega_n;\underline{\sigma_k})}{1 + \sum_{m=1}^{M} SNR_m(\omega_n)} \tag{22}$$

and where $SNR_m(\omega_n)$ is given by (19). The log likelihood (21) only depends on the differential delays $\tau_i - \tau_k$, and so we could fix one of the delays to a constant value, e.g. $\tau_M = 0$. Maximizing (21) then requires an $M-1$ dimensional search for the optimal delays $\tau_1, \ldots, \tau_{M-1}$. For $M = 2$ receivers, this is particularly easy, since we are left with the one-dimensional optimization:

$$\hat{\tau}_1 \leftarrow \max_{\tau_1} \mathrm{Re}\left[\sum_{n=L}^{U} W_{12}(\omega_n) Z_1(\omega_n) Z_2^*(\omega_n) e^{j\omega_n \tau_1}\right] \tag{23}$$

This special case was developed in [10,7,12], and is called the Generalized Cross Correlation (GCC) method [12]. It yields the ML estimate of the receiver-to-receiver delay when the weighting function $W_{12}(\omega_n)$ is precisely known. Other delay estimation techniques (e.g., [3,6,17]) have the same format as (23), but use a weighting function which is chosen to optimize a different criterion (e.g., signal-to-noise ratio, detection index, etc.). These methods are expected to outperform the conventional cross-correlation method ($W_{12}(\omega) = 1$) by taking full advantage of the spectral details of the signal and the noises.

In practice, we do not have prior knowledge of the signal gains or the spectral parameters that are required to construct $W_{12}(\omega_n)$. Therefore, it has been suggested that one estimate the weighting function first, using parametric or non-parametric spectral estimation techniques, and then use it in (23) (e.g., [7,12,3,4,9]). However, this approach is only suboptimal, and its inherent performance critically depends on the method employed for spectral estimation.

When there are $M > 2$ receivers, an approach that avoids the $(M-1)$ dimensional optimization consists of maximizing $M(M-1)/2$ separate GCC's between all possible pairs of receiver outputs to form unconstrained ML estimates of the various differential delays, $\tau_{ik} = \tau_i - \tau_k$. A weighted linear least squares fit is then used to convert these estimates into estimates of the $(M-1)$ linearly independent delays (see ([8])). This approach is asymptotically equivalent to maximizing (21). Unfortunately, $M(M-1)/2$ different GCC optimizations must be solved separately.

The chief difficulty with the ML approach to time delay estimation is that when the signal gains and spectral parameters are unknown, it requires a difficult nonlinear optimization over a large set of parameters to obtain asymptotically efficient estimates. In the next section, we will develop a general class of iterative algorithms for solving ML problems. These algorithms effectively decouple the estimation of each of the unknowns, allowing us to solve independent optimizations for each of the parameters. This computational simplification is achieved without any loss of estimation accuracy. The algorithms converge to a local maximum or stationary point of the likelihood function, increasing the likelihood on each step, and we can compute their convergence rates analytically. Furthermore, the methods naturally generate estimates of the underlying signal $s(t)$ and its variance as they iterate, which is helpful if one of the goals is target identification or analysis.

3 Iterative Likelihood Maximization Algorithms

In this section we will derive a family of iterative algorithms for solving Maximum Likelihood problems. Let $\underline{z}$ be a (finite) vector of observations, let $\underline{\xi}$ be a (finite) vector of parameters, and let $\underline{s}$ be a (finite) set of "internal" signals which cannot be directly observed, but which are stochastically related to $\underline{z}$. The ML problem we would like to solve is:

$$\hat{\underline{\xi}}_{ML} \leftarrow \max_{\underline{\xi}} \log p\left(\underline{z}; \underline{\xi}\right) \tag{24}$$

where $p\left(\underline{z}; \underline{\xi}\right)$ denotes the probability density of $\underline{z}$ given the parameter values $\underline{\xi}$. We will define $L_Z(\underline{\xi}) = \log p\left(\underline{z}; \underline{\xi}\right)$, the log likelihood of the observed data.

3.1 Estimate-Maximize-like Algorithms

We will solve this problem indirectly, using an approach first presented in [16,15]. One version of the resulting algorithm is equivalent to the Estimate-Maximize algorithm of[5]. Our derivation, however, is quite different. It yields considerable insight into the hill-climbing behavior of the algorithm, and also allows us to develop better algorithms with faster convergence rates.

The key idea is that if we knew the noise free signal value $\underline{s}$, then we could estimate the parameters by solving an ML problem which is often simpler:

$$\underline{\xi}_{ML} \leftarrow \max_{\underline{\xi}} \log\ p\left(\underline{s}, \underline{z} \middle| \underline{\xi}\right) \tag{25}$$

Unfortunately the signal $\underline{s}$ is unknown. The algorithms we will introduce bypass this difficulty by estimating the signal $\underline{s}$ using the available observations $\underline{z}$, together with the current parameter estimate $\hat{\underline{\xi}}^{(l)}$. Using this signal estimate, possibly together with its variance or other higher order moments, we reestimate the model parameters by solving a problem similar to (25). On the next iteration, the improved parameter estimates are used to further improve the signal estimate, which in turn will lead to even better parameter estimates. We will show that each iteration cycle increases the observed data likelihood function $L_Z(\underline{\xi})$ in (24), and convergence to a stationary point of $L_Z(\underline{\xi})$ is guaranteed.

To derive the algorithm, let $\tilde{\underline{\xi}}$ be a second set of parameter values. Now define:

$$Q(\tilde{\underline{\xi}};\underline{\xi}) = \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right) \log \frac{p\left(\underline{s},\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)}\, d\underline{s} \tag{26}$$

where $p\left(\underline{s},\underline{z};\underline{\xi}\right)$ is the joint probability density of $\underline{s}$ and $\underline{z}$ given parameter values $\underline{\xi}$, and $p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)$ is the conditional density of $\underline{s}$ given the observations $\underline{z}$ with parameter values $\tilde{\underline{\xi}}$. Using some algebra, we can rewrite $Q(\tilde{\underline{\xi}};\underline{\xi})$ in the form:

$$\begin{aligned}
Q(\tilde{\underline{\xi}};\underline{\xi}) &= \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)\left[\log \frac{p\left(\underline{s},\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\underline{\xi}\right)} + \log \frac{p\left(\underline{s}\,\middle|\,\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)}\right] d\underline{s} \\
&= \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)\left[\log p\left(\underline{z};\underline{\xi}\right) + \log \frac{p\left(\underline{s}\,\middle|\,\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)}\right] d\underline{s} \\
&= \log p\left(\underline{z};\underline{\xi}\right) \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right) d\underline{s} + \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right) \log \frac{p\left(\underline{s}\,\middle|\,\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)} d\underline{s} \\
&= \log p\left(\underline{z};\underline{\xi}\right) + \int p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right) \log \frac{p\left(\underline{s}\,\middle|\,\underline{z};\underline{\xi}\right)}{p\left(\underline{s}\,\middle|\,\underline{z};\tilde{\underline{\xi}}\right)} d\underline{s}
\end{aligned} \tag{27}$$

where in the transition from the first line to the second line we have used Bayes' Rule.

Now let us consider the problem of optimizing $Q(\tilde{\underline{\xi}};\underline{\xi})$. Maximizing over $\tilde{\underline{\xi}}$, Jensen's inequality applied to (27) implies that the maximum is achieved at $\tilde{\underline{\xi}} = \underline{\xi}$. Substituting back into (27):

$$\max_{\tilde{\underline{\xi}}} Q(\tilde{\underline{\xi}};\underline{\xi}) = Q(\underline{\xi};\underline{\xi}) = \log \mathrm{p}\left(\underline{z};\underline{\xi}\right) = L_Z(\underline{\xi}) \tag{28}$$

The global maximum of Q must therefore occur at $\tilde{\underline{\xi}} = \underline{\xi} = \hat{\underline{\xi}}_{ML}$. This implies that any algorithm which maximizes Q jointly over $\tilde{\underline{\xi}}$ and $\underline{\xi}$ will also maximize the likelihood function $L_Z(\underline{\xi})$.

Consider, for example, the following iterative coordinate ascent approach for maximizing Q:

$$\begin{aligned}
&\text{Guess } \hat{\underline{\xi}}^{(0)} \\
&\text{For } l = 0, 1, 2, \ldots \\
&\qquad \hat{\tilde{\underline{\xi}}}^{(l+1)} \leftarrow \max_{\tilde{\underline{\xi}}} Q\left(\tilde{\underline{\xi}};\hat{\underline{\xi}}^{(l)}\right) \\
&\qquad \hat{\underline{\xi}}^{(l+1)} \leftarrow \max_{\underline{\xi}} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)};\underline{\xi}\right)
\end{aligned} \tag{29}$$

By Jensen's inequality, the first step above yields:

$$\hat{\underline{\xi}}^{(l+1)} = \hat{\underline{\xi}}^{(l)} \tag{30}$$

Thus:

$$\hat{\underline{\xi}}^{(l+1)} \leftarrow \max_{\underline{\xi}} Q\left(\hat{\underline{\xi}}^{(l)}; \underline{\xi}\right) \tag{31}$$

To solve this, we use formula (26). Omitting terms which do not depend on $\underline{\xi}$, we get Dempster's EM algorithm [5]:

For $l = 0, 1, 2, \ldots$

$$\hat{\underline{\xi}}^{(l+1)} \leftarrow \max_{\underline{\xi}} \int p\left(\underline{s} \left| \underline{z}; \tilde{\underline{\xi}}\right.\right) \log p\left(\underline{s}, \underline{z}; \underline{\xi}\right) \, d\underline{s} \tag{32}$$

$$\tag{33}$$

or equivalently:

$$\hat{\underline{\xi}}^{(l+1)} \leftarrow \max_{\underline{\xi}} \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[L_Y\left(\underline{\xi}\right) \middle| \underline{z}\right] \tag{34}$$

where $\underline{y} = \left(\underline{s}^T \; \underline{z}^T\right)^T$ is termed the *complete data*, and where L_Y is the log likelihood of the complete data $\underline{y}$:

$$L_Y\left(\underline{\xi}\right) = \log p\left(\underline{s}, \underline{z}; \underline{\xi}\right) \tag{35}$$

The notation $\mathrm{E}_{\hat{\underline{\xi}}^{(l)}}[\ldots|\underline{z}]$ denotes the conditional expectation with respect to the observations $\underline{z}$ and with respect to parameter values $\hat{\underline{\xi}}^{(l)}$.

The computation in the EM algorithm naturally divides into two phases. The "Estimate" step (E-step) uses the latest parameter estimates $\hat{\underline{\xi}}^{(l)}$ to compute the expected value of the signal, possibly together with its variance and higher-order moments, in order to evaluate the expected value of $L_Y(\underline{\xi})$. The "Maximize" step (M-step) then maximizes over $\underline{\xi}$ to get a better parameter estimate. On the next iteration, we use the improved parameter estimate $\hat{\underline{\xi}}^{(l+1)}$ to improve the expectation calculation in the next E-step, and thereby improve the next parameter estimates in the next M-step. Clearly, each iteration increases the value of Q. Therefore,

$$\begin{aligned} L_Z\left(\hat{\underline{\xi}}^{(l+1)}\right) &= Q\left(\hat{\underline{\xi}}^{(l+1)}; \hat{\underline{\xi}}^{(l+1)}\right) \\ &\geq Q\left(\hat{\underline{\xi}}^{(l)}; \hat{\underline{\xi}}^{(l)}\right) = L_Z\left(\hat{\underline{\xi}}^{(l)}\right) \end{aligned} \tag{36}$$

Thus each iteration increases the log likelihood. In view of (29), the EM algorithm is simply a coordinate ascent optimization applied to $Q()$. Therefore, if $Q(\tilde{\underline{\xi}};\underline{\xi})$ is bounded above and differentiable in $\tilde{\underline{\xi}}$ and $\underline{\xi}$, and the estimates $\hat{\underline{\xi}}^{(l)}$ remain within a compact set, then it can be shown [16,15,5,20] that the EM algorithm is guaranteed to converge to the set of local maxima and stationary points of the observed data likelihood function $L_Z(\underline{\xi})$. Of course, as in all "hill-climbing" algorithms, the convergence point may not be the global maximum of the objective function, and thus several starting points or an initial coarse grid search may be needed to locate the global maximum.

3.2 Iterative Hybrid EM-ML Algorithms

As indicated earlier, any algorithm which iteratively maximizes Q will also iteratively maximize the likelihood function. For example, we can derive an iterative algorithm which we call "EM-ML" by using a different coordinate ascent procedure. Partition $\underline{\xi}$ into two disjoint sets of parameters, $\underline{\xi} = (\underline{\xi}_1, \underline{\xi}_2)$. Partition $\tilde{\underline{\xi}}$ similarly, $\tilde{\underline{\xi}} = (\tilde{\underline{\xi}}_1, \tilde{\underline{\xi}}_2)$. Now consider the following approach for optimizing $Q(\tilde{\underline{\xi}};\underline{\xi})$:

$$\begin{aligned} &\text{For } l = 0, 1, 2, \ldots \\ &\qquad \hat{\tilde{\underline{\xi}}}^{(l+1)}, \hat{\underline{\xi}}_1^{(l+1)} \leftarrow \max_{\tilde{\underline{\xi}}, \underline{\xi}_1} Q\left(\tilde{\underline{\xi}}; \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) \\ &\qquad \hat{\underline{\xi}}_2^{(l+1)} \leftarrow \max_{\underline{\xi}_2} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \hat{\underline{\xi}}_1^{(l+1)}, \underline{\xi}_2\right) \end{aligned} \tag{37}$$

This is similar to the hill climbing approach used to derive EM, but the order in which we search through the parameters is different.

Now to solve the first step, note that:

$$\max_{\tilde{\underline{\xi}}, \underline{\xi}_1} Q\left(\tilde{\underline{\xi}}; \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) = \max_{\underline{\xi}_1} \left\{ \max_{\tilde{\underline{\xi}}} Q\left(\tilde{\underline{\xi}}; \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) \right\} \tag{38}$$

Because of Jensen's inequality, the maximum inside the brackets is achieved at $(\hat{\tilde{\underline{\xi}}}_1^{(l+1)}, \hat{\underline{\xi}}_2^{(l+1)})$ $= (\underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)})$. Substituting back and maximizing over $\underline{\xi}_1$ gives:

$$\begin{aligned} \hat{\underline{\xi}}_1^{(l+1)} &\leftarrow \max_{\underline{\xi}_1} Q\left(\underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}; \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) \\ &= \max_{\underline{\xi}_1} \log p\left(\underline{z} \middle| \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) \end{aligned} \tag{39}$$

Thus the first set of parameters $\underline{\xi}_1$ is estimated by an ML-like step.

Solving the second step in (37) gives an EM-like step:

$$\hat{\underline{\xi}}_2^{(l+1)} \leftarrow \max_{\underline{\xi}_2} \mathrm{E}_{(\hat{\underline{\xi}}_1^{(l+1)}, \hat{\underline{\xi}}_2^{(l)})} \left[\log p \left(\underline{s}, \underline{z} \left| \hat{\underline{\xi}}_1^{(l+1)}, \underline{\xi}_2 \right. \right) \middle| \underline{z} \right] \tag{40}$$

Each iteration clearly increases the value of $Q(\hat{\underline{\xi}}^{(l)}; \hat{\underline{\xi}}^{(l)})$. By the same argument used in (36), each iteration must also increase $L_Z(\underline{\xi})$. Furthermore, under the same regularity conditions as for the EM algorithm, it can be shown that the EM-ML algorithm will converge to the set of stationary points of Q, which in turn are just the stationary points of $L_Z(\underline{\xi})$.

The name "EM-ML" is intended to reflect the use of direct ML optimization for one set of parameters, $\underline{\xi}_1$, and EM optimization for the remainder, $\underline{\xi}_2$. Obviously, there are many possible variations to this approach. We may partition $\underline{\xi}$ into more than 2 subsets, and we may maximize Q with respect to any combination of parameters in any sequence we desire. Different optimization approaches will have different convergence rates and different computational burdens. However, as long as we maximize Q with respect to every parameter at least once per iteration, we will still converge to the set of stationary points of Q and of $L_Z(\underline{\xi})$.

4 EM-like Time Delay Estimation Algorithms

Consider now applying an EM-like algorithm to the time delay estimation problem. Let $\underline{z}$ be the vector of samples of the Fourier transform of the observed data defined in (8). A natural choice for $\underline{s}$ is the vector of Fourier transform coefficients $S(\omega_n)$ of the unobserved (internal) source signal $s(t)$:

$$\underline{s} = \begin{pmatrix} S(\omega_L) \\ \vdots \\ S(\omega_U) \end{pmatrix} \tag{41}$$

The complete data $\underline{y}$ is just the set of all the components in $\underline{s}$ and $\underline{z}$. Since $s(t)$, $v(t)$, and $z_i(t)$ for $i = 1, 2, \ldots, M$ are jointly Gaussian processes, then for $WT/2\pi \gg 1$ the samples $S(\omega_n)$, $\underline{V}(\omega_n)$, and $\underline{Z}(\omega_n)$ are Gaussian variables which are statistically independent of any other samples $S(\omega_m)$, $\underline{V}(\omega_m)$, or $\underline{Z}(\omega_m)$ for $m \neq n$. Therefore the log likelihood of the complete data is given by:

$$\begin{aligned} L_Y\left(\underline{\xi}\right) &= \log p\left(S(\omega_L), \underline{Z}(\omega_L), \cdots, S(\omega_U), \underline{Z}(\omega_U); \underline{\xi}\right) \\ &= \sum_{n=L}^{U} \log p\left(S(\omega_n), \underline{Z}(\omega_n); \underline{\xi}\right) \\ &= \sum_{n=L}^{U} \left[\log p\left(S(\omega_n); \underline{\xi}\right) + \sum_{i=1}^{M} \log p\left(Z_i(\omega_n) \,\middle|\, S(\omega_n); \underline{\xi}\right)\right] \end{aligned} \tag{42}$$

where:

$$\log p\left(S(\omega_n); \underline{\xi}\right) = -\left[\log\left(\pi P_S(\omega_n; \underline{\theta})\right) - \frac{|S(\omega_n)|^2}{P_S(\omega_n; \underline{\theta})}\right] \tag{43}$$

$$\log p\left(Z_i(\omega_n) \,\middle|\, S(\omega_n); \underline{\xi}\right) = -\left[\log\left(\pi P_{V_i}(\omega_n; \underline{\sigma}_i)\right) - \frac{\left|Z_i(\omega_n) - \alpha_i e^{-j\omega_n \tau_i} S(\omega_n)\right|^2}{P_{V_i}(\omega_n; \underline{\sigma}_i)}\right] \tag{44}$$

We are now ready to apply one of our iterative algorithms. We could try using the EM algorithm described in the previous section, but in general this approach does not decouple the parameter estimation steps quite as effectively as the following approach. Partition the parameters into two sets, $\underline{\xi}_1 = (\underline{\tau}, \underline{\alpha}, \underline{\theta})$ and $\underline{\xi}_2 = (\underline{\sigma})$. Now use the following coordinate ascent algorithm to iteratively maximize $Q(\underline{\tilde{\xi}}; \underline{\xi})$:

For $l = 0, 1, 2, \ldots$

$$
\begin{aligned}
\hat{\tilde{\underline{\xi}}}^{(l+1)} &\leftarrow \max_{\tilde{\underline{\xi}}} Q\left(\tilde{\underline{\xi}}; \hat{\underline{\xi}}_1^{(l)}, \hat{\underline{\xi}}_2^{(l)}\right) \\
\hat{\underline{\xi}}_1^{(l+1)} &\leftarrow \max_{\underline{\xi}_1} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \underline{\xi}_1, \hat{\underline{\xi}}_2^{(l)}\right) \\
\hat{\underline{\xi}}_2^{(l+1)} &\leftarrow \max_{\underline{\xi}_2} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \hat{\underline{\xi}}_1^{(l+1)}, \underline{\xi}_2\right)
\end{aligned} \tag{45}
$$

As before, by Jensen's inequality the solution in the first step is $\hat{\tilde{\underline{\xi}}}^{(l+1)} = (\hat{\underline{\xi}}_1^{(l)}, \hat{\underline{\xi}}_2^{(l)})$. Substituting back gives:

For $l = 0, 1, 2, \ldots$

$$
\hat{\underline{\tau}}^{(l+1)}, \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\theta}}^{(l+1)} \leftarrow \max_{\underline{\tau},\underline{\alpha},\underline{\theta}} \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[L_Y\left(\underline{\tau}, \underline{\alpha}, \underline{\theta}, \hat{\underline{\sigma}}^{(l)}\right) \middle| \underline{z}\right] \tag{46}
$$

$$
\hat{\underline{\sigma}}^{(l+1)} \leftarrow \max_{\underline{\sigma}} \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[L_Y\left(\hat{\underline{\tau}}^{(l+1)}, \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\theta}}^{(l+1)}, \underline{\sigma}\right) \middle| \underline{z}\right] \tag{47}
$$

To solve this, substitute (43) and (44) into (42) and take the conditional expectation given $\underline{z}$ at the parameter estimate $\hat{\underline{\xi}}^{(l)}$:

$$
\begin{aligned}
\mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[L_Y\left(\underline{\xi}\right)|\underline{z}\right] &= -\left[\sum_{n=L}^{U} \log \pi P_S(\omega_n;\underline{\theta}) + \frac{|\widehat{S(\omega_n)}|^{2^{(l)}}}{P_S(\omega_n;\underline{\theta})}\right] \\
&\quad + \sum_{i=1}^{M}\left\{2\alpha_i \mathrm{Re}\left[\sum_{n=L}^{U} e^{-j\omega_n\tau_i}\hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n)/P_{V_i}(\omega_n;\underline{\sigma}_i)\right]\right. \\
&\quad - \sum_{n=L}^{U}\frac{|Z_i(\omega_n)|^2}{P_{V_i}(\omega_n;\underline{\sigma}_i)} - \alpha_i^2\sum_{n=L}^{U}\frac{|\widehat{S(\omega_n)}|^{2^{(l)}}}{P_{V_i}(\omega_n;\underline{\sigma}_i)} \\
&\quad \left. - \sum_{n=L}^{U}\log \pi P_{V_i}(\omega_n;\underline{\sigma}_i)\right\}
\end{aligned} \tag{48}
$$

where:

$$
\begin{aligned}
\hat{S}^{(l)}(\omega_n) &= \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[S(\omega_n)|\underline{z}\right] \\
&= \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[S(\omega_n)|\underline{Z}(\omega_1), \underline{Z}(\omega_2)\ldots, \underline{Z}(\omega_N)\right]
\end{aligned} \tag{49}
$$

$$
\begin{aligned}
|\widehat{S(\omega_n)}|^{2^{(l)}} &= \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[|S(\omega_n)|^2|\underline{z}\right] \\
&= \mathrm{E}_{\hat{\underline{\xi}}^{(l)}}\left[S(\omega_n)|\underline{Z}(\omega_1), \underline{Z}(\omega_2), \ldots, \underline{Z}(\omega_N)\right]
\end{aligned} \tag{50}
$$

The terms in (49) and (50) are the only conditional expectations required in this iterative algorithm. For large $WT/2\pi$, $(S(\omega_n), \underline{Z}(\omega_n))$ are statistically independent of $\underline{Z}(\omega_m)$ for all frequencies $m \neq n$. Thus:

$$\hat{S}^{(l)}(\omega_n) = \mathrm{E}_{\underline{\hat{\xi}}^{(l)}}\left[S(\omega_n)\,|\underline{Z}(\omega_n)\right] \tag{51}$$

$$\widehat{|S(\omega_n)|^2}^{(l)} = \mathrm{E}_{\underline{\hat{\xi}}^{(l)}}\left[|S(\omega_n)|^2\,|\underline{Z}(\omega_n)\right] \tag{52}$$

Since $S(\omega_n)$ and $\underline{Z}(\omega_n)$ are jointly Gaussian, invoking well-known results (e.g. [2, chap. 2]):

$$\hat{S}^{(l)}(\omega_n) = \mathbf{P}_{SZ}^{(l)}(\omega_n)\left[\mathbf{P}_{Z}^{(l)}(\omega_n)\right]^{-1}\underline{Z}(\omega_n) \tag{53}$$

$$\widehat{|S(\omega_n)|^2}^{(l)} = \left|\hat{S}^{(l)}(\omega_n)\right|^2 + \mathrm{Var}_{\underline{\hat{\xi}}^{(l)}}\left[S(\omega_n)\,|\underline{Z}(\omega_n)\right] \tag{54}$$

$$\mathrm{Var}_{\underline{\hat{\xi}}^{(l)}}\left[S(\omega_n)\,|\underline{Z}(\omega_n)\right] = P_S(\omega_n;\underline{\hat{\theta}}^{(l)}) - \mathbf{P}_{SZ}^{(l)}(\omega_n)\left[\mathbf{P}_{Z}^{(l)}(\omega_n)\right]^{-1}\mathbf{P}_{ZS}^{(l)}(\omega_n) \tag{55}$$

where from (6):

$$\begin{aligned}\mathbf{P}_{SZ}^{(l)}(\omega_n) &= \left[\mathbf{P}_{ZS}^{(l)}(\omega_n)\right]^* = \mathrm{E}_{\underline{\hat{\xi}}^{(l)}}\left[S(\omega_n)\underline{Z}^*(\omega_n)\right] \\ &= P_S(\omega_n;\underline{\hat{\theta}}^{(l)})\underline{U}^*\left(\omega_n;\underline{\hat{\tau}}^{(l)},\underline{\hat{\alpha}}^{(l)}\right)\end{aligned} \tag{56}$$

and:

$$\begin{aligned}\mathbf{P}_{Z}^{(l)}(\omega_n) &= \mathrm{E}_{\underline{\hat{\xi}}^{(l)}}\left[\underline{Z}(\omega_n)\underline{Z}^*(\omega_n)\right] \\ &= \underline{U}\left(\omega_n;\underline{\hat{\tau}}^{(l)},\underline{\hat{\alpha}}^{(l)}\right)P_S(\omega_n;\underline{\hat{\theta}}^{(l)})\underline{U}^*\left(\omega_n;\underline{\hat{\tau}}^{(l)},\underline{\hat{\alpha}}^{(l)}\right) + \mathbf{P}_v(\omega_n;\underline{\sigma})\end{aligned} \tag{57}$$

Substituting (56,57) into (53,54) and simplifying gives:

E-step: Compute:

$$\hat{S}^{(l)}(\omega_n) = \sum_{i=1}^{M}\hat{\alpha}_i^{(l)}e^{j\omega_n\hat{\tau}_i^{(l)}}Z_i(\omega_n)\hat{T}^{(l)}(\omega_n)/P_{V_i}(\omega_n;\underline{\hat{\sigma}}_i^{(l)}) \tag{58}$$

$$\widehat{|S(\omega_n)|^2}^{(l)} = \left|\hat{S}^{(l)}(\omega_n)\right|^2 + \hat{T}^{(l)}(\omega_n) \tag{59}$$

$$\text{where } \hat{T}^{(l)}(\omega_n) = \frac{P_S(\omega_n;\underline{\hat{\theta}}^{(l)})}{1+\sum_{k=1}^{M}\widehat{SNR}_k^{(l)}(\omega_n)} \tag{60}$$

and where $\widehat{SNR}_k^{(l)}(\omega_n)$ has the same formula as (19), but with the parameter values set to the current estimates:

$$\widehat{SNR}_k^{(l)}(\omega_n) = \frac{\hat{\alpha}_k^{(l)^2} P_S(\omega_n; \hat{\underline{\theta}}^{(l)})}{P_{V_k}(\omega_n; \hat{\underline{\sigma}}_k^{(l)})} \tag{61}$$

The M-step following the E-step requires maximizing (48) first with respect to $\underline{\tau}$, $\underline{\alpha}$, and $\underline{\theta}$, and then with respect to $\underline{\sigma}$. We observe that the first term in (48) depends only on $\underline{\theta}$. We further observe that the i^{th} component in the sum composing the remaining terms depends only on the i^{th} channel's delay, gain, and noise spectrum parameters τ_i, α_i, and $\underline{\sigma}_i$. Because these terms are quadratic in α_i, we get a closed form solution for the optimal choice of the α_i's. Substituting back, the maximization over the unknown parameters decouples as follows:

M-Step:

$$\hat{\underline{\theta}}^{(l+1)} \leftarrow \min_{\underline{\theta}} \sum_{n=L}^{U} \left[\log P_S(\omega_n; \underline{\theta}) + \frac{|\widehat{S(\omega_n)}|^{2^{(l)}}}{P_S(\omega_n; \underline{\theta})} \right] \tag{62}$$

For $i = 1, 2, \ldots, M$ solve :

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \mathrm{Re} \left[\sum_{n=L}^{U} e^{-j\omega_n \tau_i} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n) / P_{V_i}(\omega_n; \hat{\underline{\sigma}}_i^{(l)}) \right] \right| \tag{63}$$

$$\hat{\alpha}_i^{(l+1)} = \frac{\mathrm{Re} \left[\sum_{n=L}^{U} e^{-j\omega_n \hat{\tau}_i^{(l+1)}} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n) / P_{V_i}(\omega_n; \hat{\underline{\sigma}}_i^{(l)}) \right]}{\sum_{n=L}^{U} |\widehat{S(\omega_n)}|^{2^{(l)}} / P_{V_i}(\omega_n; \hat{\underline{\sigma}}_i^{(l)})} \tag{64}$$

$$\hat{\underline{\sigma}}_i^{(l+1)} \leftarrow \min_{\underline{\sigma}} \sum_{n=L}^{U} \left\{ \frac{\left| Z_i(\omega_n) - \hat{\alpha}_i^{(l+1)} e^{-j\omega_n \hat{\tau}_i^{(l+1)}} \hat{S}^{(l)}(\omega_n) \right|^2 + \hat{\alpha}_i^{(l+1)^2} \hat{T}^{(l)}(\omega_n)}{P_{V_i}(\omega_n; \underline{\sigma}_i)} \right.$$
$$\left. + \log P_{V_i}(\omega_n; \underline{\sigma}_i) \right\} \tag{65}$$

If some of the parameters are known (for example, if we have set $\tau_M = 0$ or $\alpha_M = 1$) then these would not be estimated. (If we already know the value of some α_i, then the absolute value in (63) should be removed, and the expression multiplied by α_i instead.) (If we wish to constrain the channel gains to be positive, $\alpha_i \geq 0$, then remove the absolute value in (63), and if the expression in (64) is negative, set $\hat{\alpha}_i^{(l+1)} = 0$ instead.)

Both the E-step and the M-step have structures which are easy to interpret. The quantities $\hat{S}^{(l)}(\omega_n)$ and $|\widehat{S(\omega_n)}|^{2^{(l)}}$ computed in the E-step can be interpreted as the

best, minimum mean square error estimates of $S(\omega_n)$ and $|S(\omega_n)|^2$ respectively, based on the current parameter estimates and the noisy observations. The signal is formed from a weighted linear combination of delayed receiver outputs. $\widehat{|S(\omega_n)|^2}^{(l)}$ is found by squaring $\hat{S}^{(l)}(\omega_n)$, then adding the signal variance $\hat{T}^{(l)}(\omega_n)$ to compensate for the uncertainty in the signal estimate. The formulae (58,59) are, in fact, the non-causal Wiener filter applied to the M-channel data.

In the M-step, the optimization for $\underline{\hat{\theta}}^{(l+1)}$ in (62) simply fits the signal spectral density to the estimated signal periodogram $\widehat{|S(\omega_n)|^2}^{(l)}$. Each delay $\hat{\tau}_i^{(l+1)}$ is estimated by maximizing a cross-correlation between the signal estimate and the i^{th} receiver output, weighted by the inverse of the current noise spectrum estimate. The gain $\hat{\alpha}_i^{(l+1)}$ is estimated as the normalized height of the cross-correlation peak. To better understand the formula for the noise spectrum parameter estimates $\hat{\sigma}_i^{(l+1)}$, define a noise estimate $\tilde{V}_i(\omega_n) = Z_i(\omega_n) - \hat{\alpha}_i^{(l+1)} e^{-j\omega_n \hat{\tau}_i^{(l+1)}} S(\omega_n)$. Then it is easy to show that (65) can be written:

$$\underline{\hat{\sigma}}_i^{(l)} \leftarrow \min_{\underline{\sigma}_i} \sum_{n=L}^{U} \left[\frac{\mathrm{E}_{\hat{\xi}^{(l)}} \left[\left| \tilde{V}_i(\omega_n) \right|^2 \middle| \underline{z} \right]}{P_{V_i}(\omega_n; \underline{\sigma}_i)} + \log P_{V_i}(\omega_n; \underline{\sigma}_i) \right] \tag{66}$$

Thus $\underline{\sigma}_i$ is estimated by fitting the noise power spectrum $P_{V_i}(\omega_n; \underline{\sigma}_i)$ to the expected periodogram of the estimated noise, $\tilde{V}_i(\omega_n)$. All these optimizations in the M-step are similar to the solutions to the ML problem in (25) for estimating the parameters given both the signal $\underline{s}$ and the observations $\underline{z}$. The difference is that the sufficient statistics $S(\omega_n)$ and $|S(\omega_n)|^2$ are replaced by their current estimates $\hat{S}^{(l)}(\omega_n)$ and $\widehat{|S(\omega_n)|^2}^{(l)}$ respectively.

The algorithm iterates back and forth, using the newest parameter estimates in the E-step to build a better Wiener filter, thus generating better signal and signal variance estimates. These in turn are used to further improve the parameter estimates on the next M-step, with the signal variance used to compensate for the uncertainty in the signal estimate. Each iteration increases $Q(\underline{\tilde{\xi}}; \underline{\xi})$ and thus also increases the log likelihood of the observations, $L_Z(\underline{\xi})$ on every step. Furthermore, because the algorithm is a simple coordinate ascent approach for maximizing $Q(\underline{\tilde{\xi}}; \underline{\xi})$, it can be shown that under the appropriate regularity conditions, convergence is guaranteed to the set of

stationary points of $L_Z(\underline{\xi})$.

A computationally attractive feature of this algorithm is that it decouples the optimization of all the parameters, allowing us to independently estimate the signal spectrum parameters $\underline{\theta}$, and each channel's delay, gain, and noise spectrum parameters τ_i, α_i, and $\underline{\sigma}_i$. We have therefore replaced the full multi-dimensional search associated with the direct maximization of (14) by an iterative search in much smaller dimensional parameter sub-spaces, leading to a considerable savings in computation.

4.1 Exploiting Parameter Ambiguity

There are several ambiguities in our model which can be deliberately exploited to improve the convergence rate. One important issue is the choice of which delay parameters to estimate. As noted earlier, if the signal waveform $s(t)$ is unknown, it is only possible to estimate the relative delays $\tau_i - \tau_j$ between receiver channels. This suggests that we set one of the delays to an arbitrary value, e.g. $\tau_M = 0$. We then only need to estimate the remaining $M-1$ delays $\tau_1, \ldots, \tau_{M-1}$. Alternatively, we could estimate all M delays, and iterate until the relative differences $\hat{\tau}_i - \hat{\tau}_j$ have converged. In the latter case, the delay estimates will be offset by some constant which depends on the initial guesses. Also we will have to solve M rather than $(M-1)$ separate cross-correlation maximizations on each iteration. However, we will show later that this extra computation achieves more rapid and more reliable convergence.

A similar issue involves the estimates of the signal gain. Suppose that one of our signal spectrum model parameters, perhaps $\theta_1 = g^2$, controls the gain of the signal spectrum, while the remaining parameters, call them $\underline{\theta}_2$, control the shape: $P_S(\omega_n; \underline{\theta}) = g^2 P_S(\omega_n; \underline{\theta}_2)$. In this case, the likelihood $L_Z(\underline{\xi})$ depends only the products of the gains $g\alpha_i$, and thus only the products $g\alpha_i$ can be identified from the observation data. We could remove this ambiguity by setting one of the gains to a fixed value, e.g. $g = 1$ or $\alpha_M = 1$. Alternatively, we could try estimating all the parameters $\alpha_1, \ldots, \alpha_M, g$. The products of the estimates $\hat{g}^{(l)}\hat{\alpha}_i^{(l)}$ will converge to their ML estimates, though the final values of the individual parameters will depend on their initial guesses. As we show later, although this latter approach involves more computational effort, it achieves

much faster convergence of the signal energy estimates.

4.2 Signal and Noise Power Spectrum Models

The complexity of the optimizations over $\underline{\theta}$ and $\underline{\sigma}$ depend primarily on the structure of the signal and noise spectral models. A convenient model for the source signal is a discrete all-pole (autoregressive, or AR) process. Suppose the model has p poles. Let $\underline{\theta} = (g, \underline{a})$, where g is the model gain, and $a_1, \ldots, a_p$ are the p AR coefficients. If W is the maximum signal bandwidth, the signal spectrum model is:

$$P_S(\omega_n; \underline{\theta}) = \frac{g^2}{\left|1 + \sum_{m=1}^{p} a_m e^{j\pi\omega_n m/W}\right|^2} \tag{67}$$

Substituting this into (62), the resulting function is quadratic in the pole coefficients $\underline{a}$. Setting the derivatives with respect to g and $\underline{a}$ to zero yields a set of linear equations to be solved for the optimal signal model coefficients:

M-Step - Estimate $\hat{\underline{\theta}}^{(l+1)}$ by:

$$\text{Solve } \hat{\mathbf{R}}_S^{(l)} \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} g^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \text{for } \widehat{g^2}^{(l+1)}, \hat{a}_i^{(l+1)} \tag{68}$$

These are similar to the Yule-Walker equations, but where $\hat{\mathbf{R}}_S^{(l)}$ is a $(p+1) \times (p+1)$ Toeplitz matrix whose elements are samples of the expected signal correlation:

$$\left[\hat{\mathbf{R}}_S^{(l)}\right]_{m,k} = \hat{R}_S^{(l)}(m-k) \tag{69}$$

and where $\hat{R}_S^{(l)}(m)$ is the inverse Discrete Fourier Transform (DFT) of the expected signal periodogram, $\widehat{|S(\omega_n)|^2}^{(l)}$:

$$\hat{R}_S^{(l)}(m) = \frac{1}{N} \mathrm{Re} \sum_{n=L}^{U} \widehat{|S(\omega_n)|^2}^{(l)} e^{j\pi\omega_n m/W} \tag{70}$$

As stated in the previous section, estimating both g and all the gains $\alpha_1, \ldots, \alpha_M$ introduces an ambiguity, since only the products $g\alpha_i$ can be identified from the observations. Nevertheless, fastest convergence is achieved by estimating all these parameters on each iteration, and simply normalizing the gains afterwards.

When the AR model order p is very high, the $P_S(\omega_n;\underline{\theta})$ model will have many degrees of freedom, and its estimated shape will follow $|\widehat{S(\omega_n)}|^{2^{(l)}}$ quite closely. A simpler alternative to using high order models is to use a non-parametric approach, estimating samples $P_S(\omega_n;\underline{\theta}) = P_S(\omega_n)$ independently for all frequencies ω_n. In this case, we just set the spectral estimate to the expected value of the signal periodogram:

$$\hat{P}_S^{(l+1)}(\omega_n) = |\widehat{S(\omega_n)}|^{2^{(l)}} \tag{71}$$

As pointed out earlier, the delay estimates tend to be somewhat insensitive to spectral details of the signal and noise fields. In practice, the primary advantage of signal spectrum estimation seems to be that it helps to filter out noise energy in regions where there is no signal. Therefore, if we are primarily interested in the delay estimation, we need only choose a convenient model that captures the essential features of the signal spectral distribution, such as its bandwidth and center frequency.

In a similar way, it is convenient to choose models for the noise spectrum which are easy to estimate. In some applications, the noise spectra are known up to a constant gain, that is:

$$P_{V_i}(\omega_n;\sigma_i) = \sigma_i P_{V_i}(\omega_n) \tag{72}$$

where $P_{V_i}(\omega_n)$ are known functions of ω_n. For example, $P_{V_i}(\omega_n) = 1$ for spectrally white noise, or $P_{V_i}(\omega_n) = 1/\omega_n$ for 1/f noise. In this case, the optimization for $\hat{\sigma}_i^{(l+1)}$ in (65) can be explicitly solved:

$$\hat{\sigma}_i^{(l+1)} = \frac{1}{N}\sum_{n=L}^{U}\left\{\frac{\left|Z_i(\omega_n) - \hat{\alpha}_i^{(l+1)}e^{-j\omega_n\hat{\tau}_i^{(l+1)}}\hat{S}^{(l)}(\omega_n)\right|^2 + \hat{\alpha}_i^{(l+1)^2}\hat{T}^{(l)}(\omega_n)}{P_{V_i}(\omega_n)}\right\} \tag{73}$$

This is just the average energy in the estimated noise periodogram. It is interesting to note that with this simplified noise model (72), directly applying the EM algorithm (34) would have yielded exactly the same estimation equations as algorithm (46).

It is also possible to estimate an AR model for the noise spectrum. Yule-Walker-like equations would be solved for the AR parameters, where the Toeplitz correlation matrix would be formed using estimated noise correlations derived from the inverse DFT of the estimated noise periodogram, $\mathrm{E}_{\underline{\xi}^{(l)}}\left[\left|\tilde{V}_i(\omega_n)\right|^2\middle|\underline{z}\right]$.

5 Convergence Analysis

An important issue associated with iterative algorithms is the rate of convergence. It has been observed[14] that the EM algorithm possesses a linear (geometric) rate of convergence with a factor that depends on the particular choice of complete data. In this section, we carefully analyze the convergence properties of the algorithms developed in the previous section. Based on this analysis, we will then suggest some modifications of the algorithm that improve the convergence behavior.

5.1 Convergence of the Delay Estimates

When the length of the observation interval T is much larger than the correlation time (inverse bandwidth) of the signal and noises, i.e. $WT/2\pi \gg 1$, it is possible to apply the central limit theorem to develop approximate formulas for the convergence rate of the EM time delay estimates. Consider the updating formula (63). Substituting the value of $\hat{S}^{(l)}(\omega_n)$ from (58) into (63):

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \mathrm{Re} \sum_{n=L}^{U} \sum_{k=1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) Z_i(\omega_n) Z_k^*(\omega_n) e^{j\omega_n \left(\tau_i - \hat{\tau}_k^{(l)}\right)} \right| \tag{74}$$

where $\hat{W}_{ik}^{(l)}(\omega_n)$ has the same formula as (22), but with parameter values set to the current estimates:

$$\hat{W}_{ik}^{(l)}(\omega_n) = \frac{\hat{\alpha}_i^{(l)} \hat{\alpha}_k^{(l)} P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) / P_{V_i}(\omega_n; \hat{\underline{\sigma}}_i^{(l)}) P_{V_k}(\omega_n; \hat{\underline{\sigma}}_k^{(l)})}{1 + \sum_{m=1}^{M} \widehat{SNR}_m^{(l)}(\omega_n)} \tag{75}$$

Let $\hat{w}_{ik}^{(l)}(t)$ be the inverse Fourier Transform of $\hat{W}_{ik}^{(l)}(\omega_n)$. Parseval's theorem can then be applied to (74) to approximately express it in the time domain:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \sum_{k=1}^{M} \int_{T_I}^{T_F} \int_{T_I}^{T_F} \hat{w}_{ik}^{(l)}(t-r) z_i(r+\tau_i) z_k(t+\hat{\tau}_k^{(l)})\, dt dr \right| \tag{76}$$

This is just a sum of weighted cross-correlations between the i^{th} receiver output $z_i(t)$ with all the other receiver outputs $z_k(t)$, with relative delay $\tau_i - \hat{\tau}_k^{(l)}$. If the observation interval $T = T_F - T_I$ is much greater than any of the delays, and much greater than the correlation time $2\pi/W$, then the law of large numbers guarantees that the value

of the integral in (76) will almost always be close to the expected value of the integral conditioned on the correct (but unknown) parameter values $\underline{\bar{\xi}}$. Thus for large $WT/2\pi$, each M-step approximately solves:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \sum_{k=1}^{M} \int_{T_I}^{T_F} \int_{T_I}^{T_F} \hat{w}_{ki}^{(l)}(t-r) \mathrm{E}_{\underline{\bar{\xi}}} \left[z_i(r+\tau_i) z_k(t+\hat{\tau}_k^{(l)}) \right] dt dr \right| \tag{77}$$

Converting back to the frequency domain:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \mathrm{Re} \sum_{n=L}^{U} \sum_{k=1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) \mathrm{E}_{\underline{\bar{\xi}}} \left[Z_i(\omega_n) Z_k^*(\omega_n) \right] e^{j\omega_n \left(\tau_i - \hat{\tau}_k^{(l)} \right)} \right| \tag{78}$$

From the covariance formula in (10):

$$\mathrm{E}_{\underline{\bar{\xi}}} \left[Z_i(\omega_n) Z_k^*(\omega_n) \right] = \begin{cases} \bar{\alpha}_i \bar{\alpha}_k P_S(\omega_n; \underline{\bar{\theta}}) e^{-j\omega_n(\bar{\tau}_i - \bar{\tau}_k)} & \text{for } k \neq i \\ \bar{\alpha}_i^2 P_S(\omega_n; \underline{\bar{\theta}}) + P_{V_i}(\omega_n; \underline{\bar{\sigma}}_i) & \text{for } k = i \end{cases} \tag{79}$$

Thus:

$$\begin{aligned} \hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \Bigg| & \mathrm{Re} \sum_{n=L}^{U} \hat{W}_{ii}^{(l)}(\omega_n) P_{V_i}(\omega_n; \underline{\bar{\sigma}}_i) e^{j\omega_n \left(\tau_i - \hat{\tau}_i^{(l)} \right)} \\ & + \sum_{n=L}^{U} \sum_{k=1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) \bar{\alpha}_i \bar{\alpha}_k P_S(\omega_n; \underline{\bar{\theta}}) e^{j\omega_n \left(\tau_i - \bar{\tau}_i + \bar{\tau}_k - \hat{\tau}_k^{(l)} \right)} \Bigg| \end{aligned} \tag{80}$$

Let us define "weighted signal and noise covariance" functions $\hat{R}_{s_{ik}}^{(l)}(t)$ and $\hat{R}_{v_i}^{(l)}(t)$ as the terms in the equation above:

$$\hat{R}_{s_{ik}}^{(l)}(t) = \frac{2\pi}{T} \mathrm{Re} \sum_{n=L}^{U} \hat{W}_{ik}^{(l)}(\omega_n) \bar{\alpha}_i \bar{\alpha}_k P_S(\omega_n; \underline{\bar{\theta}}) e^{j\omega_n t} \tag{81}$$

$$\hat{R}_{v_i}^{(l)}(t) = \frac{2\pi}{T} \mathrm{Re} \sum_{n=L}^{U} \hat{W}_{ii}^{(l)}(\omega_n) P_{V_i}(\omega_n; \underline{\bar{\sigma}}_i) e^{j\omega_n t} \tag{82}$$

Note that both $\hat{R}_{s_{ik}}^{(l)}(t)$ and $\hat{R}_{v_i}^{(l)}(t)$ behave like covariance functions because they are the inverse Fourier Transforms of positive, real, symmetric functions. In particular, these functions all peak at the origin: $\hat{R}_{s_{ik}}^{(l)}(0) \geq \left| \hat{R}_{s_{ik}}^{(l)}(t) \right|$ and $\hat{R}_{v_i}^{(l)}(0) \geq \left| \hat{R}_{v_i}^{(l)}(t) \right|$ for all t. Substituting these definitions back into (80):

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \hat{R}_{v_i}^{(l)} \left(\tau_i - \hat{\tau}_i^{(l)} \right) + \sum_{k=1}^{M} \hat{R}_{s_{ik}}^{(l)} \left(\tau_i - \bar{\tau}_i + \bar{\tau}_k - \hat{\tau}_k^{(l)} \right) \right| \tag{83}$$

Note, for example, that if both $P_S(\omega_n; \underline{\hat{\theta}}^{(l)})$ and $P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})$ are flat, so that the signal and noise both behave like bandlimited white Gaussian noise over the signal frequency

band, then $\hat{W}_{ik}^{(l)}(\omega_n)$ is flat, and $\hat{w}_{ik}^{(l)}(t)$, $\hat{R}_{s_{ik}}^{(l)}(t)$, and $\hat{R}_{v_i}^{(l)}(t)$ are all differently scaled sinc functions of width $1/T$.

The function in (83) has up to M separate peaks located near delay values $\bar{\tau}_i + \left(\hat{\tau}_k^{(l)} - \bar{\tau}_k\right)$. Note also that the values of $\underline{\hat{\alpha}}^{(l)}$, $\underline{\hat{\theta}}^{(l)}$, and $\underline{\hat{\sigma}}^{(l)}$ affect $\hat{W}_{ik}^{(l)}(\omega_n)$ and thus affect the shape of the peaks of $\hat{R}_{s_{ik}}^{(l)}(t)$ and $\hat{R}_{v_i}^{(l)}(t)$, but they do not strongly affect the location of the peak of these functions, which is always at $t = 0$. Thus we would expect that the values of $\underline{\hat{\alpha}}^{(l)}$, $\underline{\hat{\theta}}^{(l)}$, and $\underline{\hat{\sigma}}^{(l)}$ will have only a moderate effect on the convergence behavior of the delay estimates in the large WT case.

5.2 Initial Convergence Behavior

Because of the multiple-peak structure of (83), the convergence behavior of the EM algorithm will be quite different depending on whether the relative delay estimates $\hat{\tau}_k^{(l)} - \hat{\tau}_i^{(l)}$ are close to their correct values $\bar{\tau}_k - \bar{\tau}_i$, or are quite far off. In this section we consider the initial convergence behavior; in the next we consider the behavior when the delay estimates are close to their final values.

To simplify the analysis, we will only consider the case of $M = 2$ receivers. We also consider two variations of the EM algorithm. In the first variation, we estimate both $\hat{\tau}_1^{(l+1)}$ and $\hat{\tau}_2^{(l+1)}$ on every iteration:

$$\hat{\tau}_1^{(l+1)} \leftarrow \max_{\tau_1} \left| \hat{R}_{s_{12}}^{(l)}\left(\tau_1 - \bar{\tau}_1 + \bar{\tau}_2 - \hat{\tau}_2^{(l)}\right) + \hat{R}_{s_{11}}^{(l)}\left(\tau_1 - \hat{\tau}_1^{(l)}\right) + \hat{R}_{v_1}^{(l)}\left(\tau_1 - \hat{\tau}_1^{(l)}\right) \right| \tag{84}$$

$$\hat{\tau}_2^{(l+1)} \leftarrow \max_{\tau_2} \left| \hat{R}_{s_{21}}^{(l)}\left(\tau_2 - \bar{\tau}_2 + \bar{\tau}_1 - \hat{\tau}_1^{(l)}\right) + \hat{R}_{s_{22}}^{(l)}\left(\tau_2 - \hat{\tau}_2^{(l)}\right) + \hat{R}_{v_2}^{(l)}\left(\tau_2 - \hat{\tau}_2^{(l)}\right) \right| \tag{85}$$

In the second variation, we set $\bar{\tau}_2 = \tau_2 = \hat{\tau}_2^{(l)} = 0$, and only estimate τ_1. In this case, we will estimate τ_1 with (84), and simply ignore the estimation equation for τ_2.

If the relative delay estimate $\hat{\tau}_2^{(l)} - \hat{\tau}_1^{(l)}$ is quite far from the correct value $\bar{\tau}_2 - \bar{\tau}_1$, then the functions being maximized in (84) and (85) will have two separate peaks, as shown in figure 1. The function in (84) has one peak at the old estimate $\hat{\tau}_1^{(l)}$, and another peak at the correct value $\bar{\tau}_1 + \left(\hat{\tau}_2^{(l)} - \bar{\tau}_2\right)$. Thus when we find the global optimum, the new estimate $\hat{\tau}_1^{(l+1)}$ will fall between the old value and the correct one, at a location which depends on the relative heights of the peaks $\hat{R}_{s_{12}}^{(l)}(0)$ and $\hat{R}_{s_{11}}^{(l)}(0) + \hat{R}_{v_1}^{(l)}(0)$. The $\hat{\tau}_2^{(l+1)}$ estimate behaves in a similar manner.

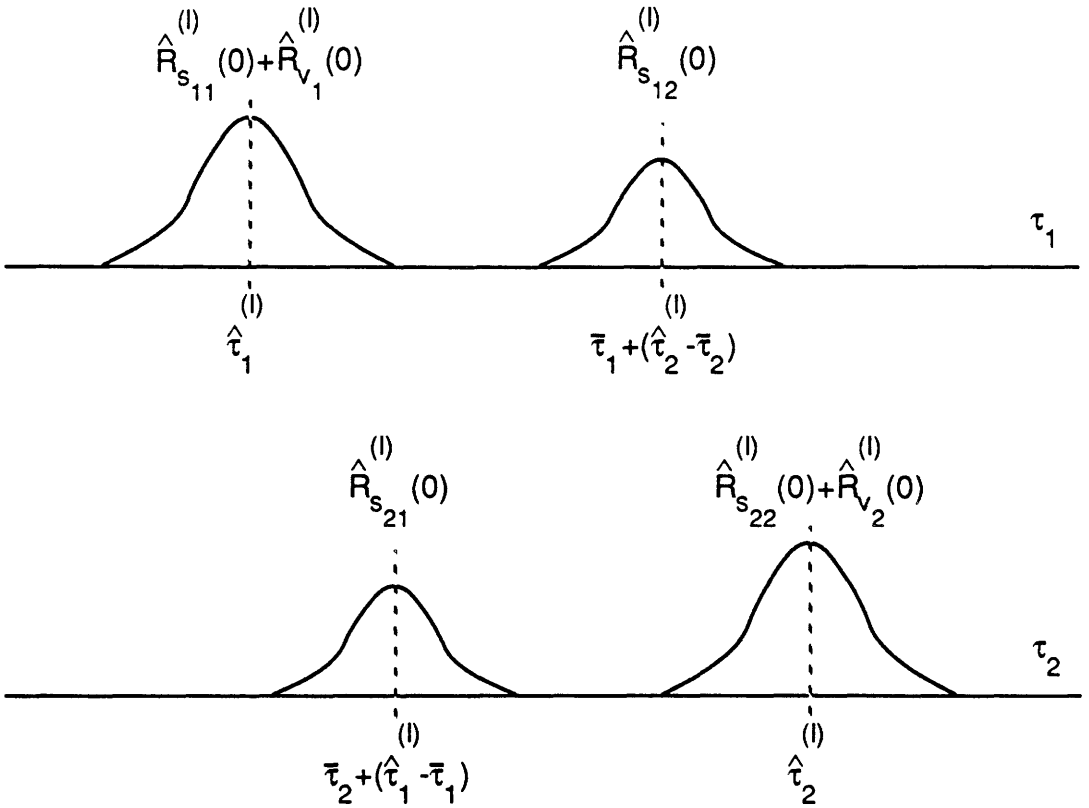


Figure 1: Functions (84) and (85) when initial estimates are far from convergence.

To gain further insight, it is convenient to consider the case where the two receivers receive similar signal and noise. Suppose $\bar{\alpha}_1 = \bar{\alpha}_2$, $\bar{\sigma}_1 = \bar{\sigma}_2$, $\hat{\sigma}_1^{(l)} = \hat{\sigma}_2^{(l)}$, and $P_{v_1}(\omega_n; \hat{\sigma}_1^{(l)}) = P_{v_2}(\omega_n; \hat{\sigma}_2^{(l)})$ for all ω_n. Then all the functions $\hat{R}_{s_{ik}}^{(l)}(t)$ are identical except that they are scaled by $\hat{\alpha}_k^{(l)}\hat{\alpha}_i^{(l)}$, and all the functions $\hat{R}_{v_i}^{(l)}(t)$ are scaled by $\hat{\alpha}_i^{(l)^2}$. We can thus distinguish three cases:

1. $\hat{\alpha}_2^{(l)} \gg \hat{\alpha}_1^{(l)}$
2. $\hat{\alpha}_2^{(l)} \approx \hat{\alpha}_1^{(l)}$
3. $\hat{\alpha}_2^{(l)} \ll \hat{\alpha}_1^{(l)}$

In the first case, we will find that $\hat{\tau}_1^{(l+1)} \approx \bar{\tau}_1 + \left(\hat{\tau}_2^{(l)} - \bar{\tau}_2\right)$ while $\hat{\tau}_2^{(l+1)} \approx \hat{\tau}_2^{(l)}$. Thus the new relative delay $\hat{\tau}_2^{(l+1)} - \hat{\tau}_1^{(l+1)}$ jumps immediately to a value very close to the correct value $\bar{\tau}_2 - \bar{\tau}_1$. Similarly, in case 3, we will find that $\hat{\tau}_1^{(l+1)} \approx \hat{\tau}_1^{(l)}$, but $\hat{\tau}_2^{(l+1)} \approx \bar{\tau}_2 + \left(\hat{\tau}_1^{(l)} - \bar{\tau}_1\right)$. The new relative delay $\hat{\tau}_2^{(l+1)} - \hat{\tau}_1^{(l+1)}$ is again very close to its correct value. However, in case 2 where the estimates of the signal gains on the receivers are close to each other, then we find that $\hat{\tau}_1^{(l+1)} \approx \hat{\tau}_1^{(l)}$ and $\hat{\tau}_2^{(l+1)} \approx \hat{\tau}_2^{(l)}$, and the new relative delay estimate $\hat{\tau}_2^{(l+1)} - \hat{\tau}_1^{(l+1)}$ remains close to its previous value $\hat{\tau}_2^{(l)} - \hat{\tau}_1^{(l)}$, and far from the correct value.

Note that if we force $\tau_2 = 0$ and only estimate τ_1, then good initial convergence will only occur in the first case, where $\hat{\alpha}_2^{(l)} \gg \hat{\alpha}_1^{(l)}$. Similarly, if we force $\tau_1 = 0$ and only estimate τ_2, then good initial convergence will only occur in the third case, where $\hat{\alpha}_2^{(l)} \ll \hat{\alpha}_1^{(l)}$. This suggests that better initial convergence may be achieved by estimating both delays, even though only the difference between them has any meaning.

This poor initial convergence behavior when the signal gain estimates are similar implies that it is important to initialize the EM algorithm with a good delay estimate. There are a couple of reasonable approaches. One is to artificially force the initial gain estimates to be greatly mismatched, for example by setting $\hat{\alpha}_1^{(0)} = 0$. In other words, estimate the signal $\hat{S}^{(0)}(\omega_n)$ using only the output of the second receiver, $Z_2(\omega_n)$. This will give reasonable initial delay estimates provided that the signal-to-noise ratio on receiver 2 is at least unity.

A similar analysis for the case of $M > 2$ shows a similar effect. When the gain estimates are all similar, and the receiver outputs have the same noise level, then the initial convergence rate can be quite slow. This effect can be partially compensated for by artificially setting all $\alpha_k = 0$ except for $k = M$, so that $\hat{S}^{(0)}(\omega_n)$ depends only on $Z_M(\omega_n)$, thereby giving reasonable initial delay estimates provided that the signal-to-noise ratio on receiver M is at least unity.

5.3 Asymptotic Convergence Rate

The EM algorithm will converge at a linear rate when the relative delay estimates $\hat{\tau}_k^{(l)} - \hat{\tau}_i^{(l)}$ are close to their correct values $\bar{\tau}_k - \bar{\tau}_i$. In this case, the M peaks of the functions in (83) will merge into a single peak located at a better estimate.

Assuming that the delay estimates are close to their correct values, we can approximate the shapes of $\hat{R}_{s_{ik}}^{(l)}(t)$ and $\hat{R}_{v_i}^{(l)}(t)$ near the origin as quadratic.

$$\hat{R}_{s_{ik}}^{(l)}(t) \approx \hat{R}_{s_{ik}}^{(l)}(0) - \beta_{ik}^{(l)} t^2 \qquad \beta_{ik}^{(l)} \geq 0 \tag{86}$$

$$\hat{R}_{v_i}^{(l)}(t) \approx \hat{R}_{v_i}^{(l)}(0) - \lambda_i^{(l)} t^2 \qquad \lambda_i^{(l)} \geq 0 \tag{87}$$

(These second order Taylor series expansions will always exist because of the finite signal and noise bandwidth assumptions). Substituting into (83), we get the following approximate optimization problems for the delays:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left[\sum_{k=1}^{M} \hat{R}_{s_{ik}}^{(l)}(0) - \beta_{ik}^{(l)} \left(\tau_i - \bar{\tau}_i + \bar{\tau}_k - \hat{\tau}_k^{(l)} \right)^2 \right] + \hat{R}_{v_i}^{(l)}(0) - \hat{\lambda}_i^{(l)} \left(\tau_i - \hat{\tau}_i^{(l)} \right)^2 \tag{88}$$

This function is quadratic in τ_i. We can find the peak by setting the derivative with respect to τ_i to zero:

$$\hat{\tau}_i^{(l+1)} = \bar{\tau}_i + \sum_{k=1}^{M} \eta_{ik}^{(l)} \left(\hat{\tau}_k^{(l)} - \bar{\tau}_k \right) \tag{89}$$

where:

$$\eta_{ik}^{(l)} = \begin{cases} \dfrac{\beta_{ik}^{(l)}}{\sum_{m=1}^{M} \beta_{im}^{(l)} + \lambda_i^{(l)}} & k \neq i \\ \dfrac{\beta_{ik}^{(l)} + \lambda_i^{(l)}}{\sum_{m=1}^{M} \beta_{im}^{(l)} + \lambda_i^{(l)}} & k = i \end{cases} \tag{90}$$

Note that $0 < \hat{\eta}_{ik}^{(l)} < 1$. To simplify the analysis, let us assume that all channels have about the same signal and noise energy, so that all the $\beta_{ik}^{(l)}$ have about the same value

$\beta^{(l)}$, and all the $\lambda_i^{(l)}$ have about the same value $\lambda^{(l)}$. If we estimate all M delays on every iteration, then the new relative delay estimates will be:

$$\hat{\tau}_i^{(l+1)} - \hat{\tau}_M^{(l)} = \gamma^{(l)} \left(\hat{\tau}_i^{(l)} - \hat{\tau}_M^{(l)} \right) + \left(1 - \gamma^{(l)} \right) \left(\bar{\tau}_i - \bar{\tau}_M \right) \tag{91}$$

where:

$$\gamma^{(l)} \approx \frac{\lambda^{(l)}}{M\beta^{(l)} + \lambda^{(l)}} \tag{92}$$

The error in the relative delay estimate will tend to converge to zero at rate $\gamma^{(l)} < 1$. If the number of channels M is large, or if the SNR is high so that $\lambda^{(l)} \ll \beta^{(l)}$, then $\gamma^{(l)} \approx 0$, and the convergence rate will be very rapid.

If we choose to fix $\tau_M = 0$, however, then the convergence rate slows significantly. From (89):

$$\hat{\tau}_i^{(l+1)} - \bar{\tau}_i = \gamma^{(l)} \left(\hat{\tau}_i^{(l)} - \bar{\tau}_i \right) + \frac{\beta^{(l)}}{M\beta^{(l)} + \lambda^{(l)}} \sum_{k=1}^{M-1} \left(\hat{\tau}_k^{(l)} - \bar{\tau}_k \right) \tag{93}$$

Summing over $i = 1, \ldots, M-1$ gives:

$$\sum_{k=1}^{M-1} \left(\hat{\tau}_k^{(l+1)} - \bar{\tau}_k \right) = \left(1 - \frac{\beta^{(l)}}{M\beta^{(l)} + \lambda^{(l)}} \right) \sum_{k=1}^{M-1} \left(\hat{\tau}_k^{(l)} - \bar{\tau}_k \right) \tag{94}$$

The error in the average delay error drops by less than $(M-1)/M$ on each iteration. Clearly, estimating all M absolute delays is essential for fast convergence.

As indicated earlier, the errors of the ML delay estimates are asymptotically uncorrelated with the errors in the ML signal gain, noise gain, and spectral parameter estimates. Therefore, if we are primarily interested in the delay estimates and we are close to the point of convergence, we may consider performing a partial M-step, leaving the spectral estimates at their current values and updating only the estimates of $\underline{\tau}$. This may save some computation, with only an insignificant effect on the rate of convergence of the algorithm and the variance of the resulting delay estimates.

5.4 Convergence Rate of the Signal Spectral Parameters

The convergence rate of the spectral parameters $\hat{\underline{\theta}}^{(l)}$ can be analyzed using similar techniques. The diagonal block structure of the Fisher Information Matrix (FIM) shows that the estimates of $\underline{\tau}$ are statistically independent of those of $\underline{\theta}$. Therefore, to reduce

the complexity of the analysis, let us assume that the correct time delays, $\bar{\tau}$, are known perfectly, so that they do not need to be estimated. Also, let us assume that the noise spectra are known and identical, so that $\hat{\sigma}_k^{(l)} = \bar{\sigma}_k$, and $P_{V_k}(\omega_n; \underline{\hat{\sigma}}_k^{(l)}) = P_{V_k}(\omega_n; \underline{\bar{\sigma}}_k) = P_V(\omega_n)$.

Consider first the non-parametric approach of modeling the signal spectrum as an arbitrary unknown function, $P_S(\omega_n)$. In that case, the estimated spectrum is given by (71). Let $\bar{P}_S(\omega_n)$ be the correct (but unknown) signal power spectrum. Substitute the formula (59) for $\left|\widehat{S(\omega_n)}\right|^{2^{(l)}}$ into (71), then assume that $WT/2\pi \gg 1$ so that we can replace the right hand side with its expected value given the correct (but unknown) power spectrum. After much algebra, we get:

$$\left(\hat{P}_S^{(l+1)}(\omega_n) - \kappa^{(l)^2}\bar{P}_S(\omega_n)\right) \approx \left(1 - \hat{\beta}^{(l)^2}(\omega_n)\right)\left(\hat{P}_S^{(l)}(\omega_n) - \kappa^{(l)^2}\bar{P}_S(\omega_n)\right) \tag{95}$$

where:

$$\hat{\beta}^{(l)}(\omega_n) = \frac{\hat{P}_S(\omega_n)\sum_{k=1}^M \hat{\alpha}_k^{(l)^2}}{P_V(\omega_n) + \hat{P}_S(\omega_n)\sum_{k=1}^M \hat{\alpha}_k^{(l)^2}} \tag{96}$$

$$\kappa^{(l)} = \frac{\sum_{k=1}^M \hat{\alpha}_k^{(l)}\bar{\alpha}_k}{\sum_{k=1}^M \hat{\alpha}_k^{(l)^2}} \tag{97}$$

The shape of the spectrum estimate converges to the correct shape at rate $\left(1 - \hat{\beta}^{(l)^2}(\omega_n)\right)$, but the gain is off by $\kappa^{(l)^2}$. At frequencies with high signal-to-noise, $\hat{\beta}^{(l)}(\omega_n) \approx 1$ and the spectrum converges very quickly. In the valleys where the signal spectrum is below the noise floor, $\hat{\beta}^{(l)}(\omega_n) \approx 0$, and the convergence rate is very slow. The gain error, $\kappa^{(l)^2}$, in the spectrum estimate reflects any mismatch in overall energy level between the current signal gain estimates $\underline{\hat{\alpha}}^{(l)}$ and the actual gains $\underline{\bar{\alpha}}$. Note that near convergence, the ratio of estimated total signal power to actual total signal power is:

$$\frac{\hat{P}_S^{(l)}(\omega_n)\sum_{k=1}^M \hat{\alpha}_k^{(l)^2}}{\bar{P}_S^{(l)}(\omega_n)\sum_{k=1}^M \bar{\alpha}_k^2} \approx \frac{\left(\sum_{k=1}^M \hat{\alpha}_k^{(l)}\bar{\alpha}_k\right)^2}{\left(\sum_{k=1}^M \hat{\alpha}_k^{(l)^2}\right)\left(\sum_{i=1}^M \bar{\alpha}_i^2\right)} \tag{98}$$

By the Cauchy Schwartz inequality, the factor on the right is always less than or equal to one, with equality if and only if the relative channel gains are correct, so that $\underline{\hat{\alpha}}^{(l)}$ is proportional to $\underline{\bar{\alpha}}$. If in fact the relative gains are correct, then we must have $\underline{\hat{\alpha}}^{(l)} = \underline{\bar{\alpha}}/\kappa^{(l)}$, where $\kappa^{(l)}$ is defined above, and the error in the signal power spectrum gain

exactly cancels the error in the signal gains, so that the total estimated signal power is correct. In other cases, where the relative gains are not correct, then the estimated total signal power will somewhat underestimate the actual total signal power.

If the signal power spectrum $P_S(\omega_n;\underline{\theta})$ is modeled as a discrete all-pole function, then the convergence behavior will be far more complicated. Nevertheless, we would expect to see the same type of behavior as for the non-parametric model, particularly for large model orders p. Total signal power will be approximately correct; if the signal gain estimates are systematically off by about $1/\kappa^{(l)}$, then the gain of the power spectrum estimate will compensate by being off by about $\kappa^{(l)^2}$. Convergence will be rapid near the peaks of the power spectrum, and slow near the valleys.

5.5 Convergence of the Signal Gains

The convergence behavior of the signal gains is more complicated to analyze. In particular, we will show that the relative gains $\hat{\alpha}_j^{(l)}/\hat{\alpha}_i^{(l)}$ converge quickly, but the average gain level converges slowly. To simplify the analysis, suppose that all the delay parameter values are known exactly, $\hat{\tau}_k^{(l)} = \bar{\tau}_k$ and that all noise spectra are known and identical, $P_{V_k}(\omega_n;\hat{\underline{\sigma}}_k^{(l)}) = P_{V_k}(\omega_n;\bar{\underline{\sigma}}_k) = P_V(\omega_n)$ for all k. Then start with the formula (64), and substitute the values for $\hat{S}^{(l)}(\omega_n)$ and $|\widehat{S(\omega_n)}|^{2^{(l)}}$ from (58) and (59). In the long observation interval case, $WT/2\pi \gg 1$, we can then replace the numerator and denominator of (64) with their expected values conditioned on the correct (but unknown) parameter values. After voluminous algebra:

$$\left(\hat{\alpha}_i^{(l+1)} - \bar{\alpha}_i/\kappa_\alpha^{(l)}\right) = \gamma_\alpha^{(l)}\left(\hat{\alpha}_i^{(l)} - \bar{\alpha}_i/\kappa_\alpha^{(l)}\right) \tag{99}$$

where:

$$\gamma_\alpha^{(l)} = \frac{\sum_{n=L}^{U}\left[\left(1-\beta^{(l)}(\omega_n)\right)P_S(\omega_n;\hat{\underline{\theta}}^{(l)})\right]/P_V(\omega_n)}{D^{(l)}(\omega_n)} \tag{100}$$

$$\kappa_\alpha^{(l)} = \frac{\left(1-\gamma_\alpha^{(l)}\right)D^{(l)}(\omega_n)}{\sum_{n=L}^{U}\left[\kappa^{(l)}(\omega_n)\beta^{(l)}(\omega_n)P_S(\omega_n;\hat{\underline{\theta}}^{(l)})\right]/P_V(\omega_n)} \tag{101}$$

where:

$$D^{(l)}(\omega_n) = \sum_{n=L}^{U} \left[\kappa^{(l)^2}(\omega_n)\beta^{(l)^2}(\omega_n)\frac{P_S(\omega_n;\underline{\bar{\theta}})}{P_V(\omega_n)} + \left(1-\beta^{(l)^2}(\omega_n)\right) P_S(\omega_n;\underline{\hat{\theta}}^{(l)}) \right] / P_V(\omega_n) \tag{102}$$

The gains converge linearly at rate $\gamma_\alpha^{(l)}$ to values $\underline{\bar{\alpha}}/\kappa_\alpha^{(l)}$ which deviate uniformly from their correct values by a constant factor $\kappa_\alpha^{(l)}$. For high SNR, $\beta^{(l)}(\omega_n) \approx 1$, and thus $\gamma_\alpha^{(l)} \approx 0$. The convergence rate will therefore be quite rapid. The error in the average level of the gains, however, tends be change very slowly, particularly at high SNR. For example, suppose that all the gain estimates initially deviate by the same factor, $\hat{\alpha}_i^{(l)} = \bar{\alpha}_i/\kappa^{(l)}$. Then $\kappa_\alpha^{(l)} \approx \kappa^{(l)}$, and the new estimates $\hat{\alpha}_i^{(l+1)}$ will again deviate from their correct values by about the same amount, $\kappa^{(l)} \approx \kappa^{(l+1)}$.

Intuitively, the problem of the poor estimates of the average signal gain level can be traced to EM's use of iteration to decouple the estimation of $S(\omega_n)$ and the parameters $\underline{\xi}$. Unfortunately, the estimates of $S(\omega_n)$ and $\underline{\alpha}$ are tightly coupled. If the gain estimates $\underline{\hat{\alpha}}^{(l)}$ are too low by a factor of κ, for example, then the estimate $\hat{S}^{(l)}(\omega_n)$ will be too large by a factor of about κ. On the next M-step, this again causes the new gain estimate $\underline{\hat{\alpha}}^{(l+1)}$ to be too low by about a factor of κ. Convergence will be very slow. Note, however, that if we estimate at least the gain of the signal power spectrum, then the estimated periodogram $|\widehat{S(\omega_n)}|^{2^{(l)}}$ will be too large by about a factor of κ^2, and thus the new power spectrum $P_S(\omega_n;\underline{\hat{\theta}}^{(l+1)})$ will be too large by about a factor κ^2. The total signal gain $\underline{\hat{\alpha}}^{(l+1)^2} P_S(\omega_n;\underline{\hat{\theta}}^{(l+1)})$ will be approximately correct. Since the delay updating procedure depends on $\underline{\hat{\alpha}}^{(l+1)}$ and $\underline{\hat{\theta}}^{(l+1)}$ only through $\underline{\hat{\alpha}}^{(l+1)^2} P_S(\omega_n;\underline{\hat{\theta}}^{(l+1)})$, this implies that large but compensating errors in the signal gain and spectral estimate gain will have little effect on the convergence of the delay estimates.

This observation suggests that we should always estimate all M gains $\alpha_1,\ldots,\alpha_M$ as well as an overall signal spectral gain g^2. The relative balance of gains between channels will converge quickly, and any systematic error in the level of $\underline{\hat{\alpha}}^{(l)}$ will be quickly compensated by a corresponding inverse error in the level of $\hat{g}^{(l)}$.

6 EM-ML Algorithms

The convergence analysis of the preceding section suggests a number of ways to improve the EM algorithm. The most disturbing problem with the delay estimation is that the functions (84,85) being optimized for $\hat{\tau}_i^{(l+1)}$ have extra peaks centered at the old estimates $\hat{\tau}_i^{(l)}$. When the old estimates are far from their true values, these potentially large peaks confuse the search for the correct peak. Near convergence, they slow the convergence rate.

We shall fix this problem, together with the problem of slow linear convergence for the gain and spectral parameter estimates, by using various EM-ML algorithms to directly optimize the observed data likelihood function $L_Z(\underline{\xi})$ for some of the parameters, while using an EM-style iteration to estimate the remaining parameters. We will show that the resulting algorithms converge more quickly than the earlier algorithms, yet are still guaranteed to increase the likelihood in every iteration, and to converge to the set of stationary points of $L_Z(\underline{\xi})$.

6.1 EM-ML Joint Delay Estimation

Consider the following EM-ML-style iterative algorithm. Partition the parameters $\underline{\xi}$ into three sets, $\underline{\xi}_1 = (\underline{\tau})$, $\underline{\xi}_2 = (\underline{\alpha}, \underline{\theta})$, and $\underline{\xi}_3 = (\underline{\sigma})$. Partition $\tilde{\underline{\xi}}$ similarly. Then use the following hill-climbing approach for optimizing $Q(\tilde{\underline{\xi}}; \underline{\xi})$:

$$\begin{aligned}
&\text{For } l = 0, 1, 2, \ldots \\
&\qquad \hat{\tilde{\underline{\xi}}}^{(l+1)}, \hat{\xi}_1^{(l+1)} \leftarrow \max_{\tilde{\underline{\xi}}, \xi_1} Q\left(\tilde{\underline{\xi}}; \xi_1, \hat{\xi}_2^{(l)}, \hat{\xi}_3^{(l)}\right) \\
&\qquad \hat{\xi}_2^{(l+1)} \leftarrow \max_{\xi_2} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \hat{\xi}_1^{(l+1)}, \xi_2, \hat{\xi}_3^{(l)}\right) \\
&\qquad \hat{\xi}_3^{(l+1)} \leftarrow \max_{\xi_3} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \hat{\xi}_1^{(l+1)}, \hat{\xi}_2^{(l+1)}, \xi_3\right)
\end{aligned} \tag{103}$$

Using the same reasoning as in section 3.2, this algorithm can be shown to be equivalent to:

$$\hat{\underline{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} L_Z\left(\underline{\tau}, \hat{\underline{\alpha}}^{(l)}, \hat{\underline{\theta}}^{(l)}, \hat{\underline{\sigma}}^{(l)}\right) \tag{104}$$

$$\underline{\hat{\alpha}}^{(l+1)}, \underline{\hat{\theta}}^{(l+1)} \leftarrow \max_{\underline{\alpha},\underline{\theta}} \mathrm{E}_{\underline{\hat{\xi}}^{(l)}} \left[\log p \left(\underline{s}, \underline{z} \middle| \underline{\hat{\tau}}^{(l+1)}, \underline{\alpha}, \underline{\theta}, \underline{\hat{\sigma}}^{(l)} \right) \middle| \underline{z} \right] \tag{105}$$

$$\underline{\hat{\sigma}}^{(l+1)} \leftarrow \max_{\underline{\sigma}} \mathrm{E}_{\underline{\hat{\xi}}^{(l)}} \left[\log p \left(\underline{s}, \underline{z} \middle| \underline{\hat{\tau}}^{(l+1)}, \underline{\hat{\alpha}}^{(l+1)}, \underline{\hat{\theta}}^{(l+1)}, \underline{\sigma} \right) \middle| \underline{z} \right] \tag{106}$$

The delays are thus estimated by directly maximizing the likelihood function, while the other parameters are estimated by EM-like steps. By directly maximizing the objective function for $\underline{\hat{\tau}}^{(l+1)}$, we increase the likelihood more rapidly than does the conventional EM algorithm, and may therefore improve the rate of convergence. Like other EM-ML algorithms, each iteration must increase the observed data likelihood function $L_Z(\underline{\xi})$, and convergence to a local maxima or a stationary point of $L_Z(\underline{\xi})$ is guaranteed under mild regularity conditions.

Substituting (13) into (104) gives:

ML-step:

$$\underline{\hat{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} \mathrm{Re} \sum_{n=L}^{U} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) Z_i(\omega_n) Z_k^*(\omega_n) e^{j\omega_n(\tau_i - \tau_k)} \tag{107}$$

where the weighting function $\hat{W}_{ik}^{(l)}(\omega_n)$ is defined in (75). Note that (107) depends only on the relative delays $\tau_i - \tau_k$, and so it is insensitive to adding a fixed constant to all the delays. We can therefore set one of the delays to a fixed value, e.g. $\tau_M = 0$, and only optimize over the remaining $M - 1$ delays.

The second and third steps of the EM-ML joint delay estimation algorithm yield similar formulas for $\underline{\hat{\alpha}}^{(l+1)}$, $\underline{\hat{\theta}}^{(l+1)}$, and $\underline{\hat{\sigma}}^{(l+1)}$ as the EM algorithm:

E-step:

Estimate $\hat{S}^{(l)}(\omega_n)$ and $\left|\widehat{S(\omega_n)}\right|^{2\,(l)}$ from (58) and (59), except use the most recent delay estimates $\underline{\hat{\tau}}^{(l+1)}$.

M-step:

Estimate $\underline{\hat{\theta}}^{(l+1)}$ from (62), $\underline{\hat{\alpha}}^{(l+1)}$ from (64), and $\underline{\hat{\sigma}}^{(l+1)}$ from (65).

For large WT we can analyze the convergence rate with the techniques used before. When $WT/2\pi \gg 1$, the function maximized over $\underline{\tau}$ in (107) will be approximately equal to its expected value conditioned on the correct (but unknown) parameters $\underline{\bar{\xi}}$. Thus:

$$\underline{\hat{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} \mathrm{Re} \sum_{n=L}^{U} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) \mathrm{E}_{\underline{\bar{\xi}}} \left[Z_i(\omega_n) Z_k^*(\omega_n) \right] e^{j\omega_n(\tau_i - \tau_k)} \tag{108}$$

Recall (10):

$$\hat{\underline{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} \mathrm{Re} \sum_{n=L}^{U} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) \left[\bar{\alpha}_i \bar{\alpha}_k P_S(\omega_n; \bar{\underline{\theta}}) e^{-j\omega_n(\bar{\tau}_i - \bar{\tau}_k)} \right] e^{j\omega_n(\tau_i - \tau_k)} \tag{109}$$

Using the definition of $\hat{R}_{s_{ik}}^{(l)}(t)$ from (81):

$$\hat{\underline{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} \sum_{i=1}^{M-1} \sum_{k=i+1}^{M} \hat{R}_{s_{ik}}^{(l)} \left(\tau_i - \bar{\tau}_i + \bar{\tau}_k - \tau_k \right) \tag{110}$$

Note that this function has a global maximum at the correct differential delay estimates $\hat{\tau}_i^{(l)} - \hat{\tau}_k^{(l)} = \bar{\tau}_i - \bar{\tau}_k$ for $i, k = 1, \ldots, M$. Thus each set of delay estimates is not biased by the previous delay estimates, and we should converge to approximately the correct answer in one step, regardless of the initial delay estimate. The convergence rate, therefore, is super-linear.

Another, less direct way to explain the fast convergence of this algorithm is to compare it with the following strategy for maximizing $Q(\tilde{\underline{\xi}}; \underline{\xi})$:

$$\begin{aligned}
&\text{For } l = 0, 1, 2, \ldots \\
&\qquad \hat{\tilde{\underline{\xi}}}^{(l+1)}, \hat{\xi}_1^{(l+1/2)} \leftarrow \max_{\tilde{\underline{\xi}}, \xi_1} Q\left(\tilde{\underline{\xi}}; \xi_1, \hat{\xi}_2^{(l)}, \hat{\xi}_3^{(l)} \right) \\
&\qquad \hat{\xi}_1^{(l+1)}, \hat{\xi}_2^{(l+1)} \leftarrow \max_{\xi_1, \xi_2} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \xi_1, \xi_2, \hat{\xi}_3^{(l)} \right) \\
&\qquad \hat{\xi}_3^{(l+1)} \leftarrow \max_{\xi_3} Q\left(\hat{\tilde{\underline{\xi}}}^{(l+1)}; \hat{\xi}_1^{(l+1)}, \hat{\xi}_2^{(l+1)}, \xi_3 \right)
\end{aligned} \tag{111}$$

This is similar to the EM-ML algorithm, except that on the second step we re-estimate ξ_1 along with ξ_2. Careful examination of the formula for this second ξ_1 estimate, however, shows that as long as the signs of the gains $\hat{\alpha}_k^{(l+1)}$ are the same as the signs of $\hat{\alpha}_k^{(l)}$, then the second delay estimate must equal the first, $\hat{\xi}_1^{(l+1)} = \hat{\xi}_1^{(l+1/2)}$. Thus, except in those rare cases where the signal gain estimates change sign, this alternative strategy generates exactly the same estimates as the EM-ML joint delay estimation algorithm. On the other hand, if we compare this alternative strategy with the EM-like strategy in section 4, the only difference is that we estimate ξ_1 along with $\tilde{\underline{\xi}}$ in the first step. Because fewer constraints are imposed during this coordinate ascent, we would expect this algorithm (and thus also EM-ML) to converge faster than EM.

6.2 EM-ML With Independent Delay Estimates

The previous EM-ML joint delay estimation algorithm requires an $(M-1)$ dimensional search to compute each delay estimate. This is not necessarily convenient for $M > 2$ receivers. A strategy which would use less computation on each iteration would start with M small steps to maximize $Q(\underline{\tilde{\xi}}; \underline{\xi})$ with respect to parameters $(\underline{\tilde{\xi}}, \tau_i)$ for $i = 1, \ldots, M$. Then maximize with respect to $(\underline{\alpha}, \underline{\theta})$, and then with respect to $(\underline{\sigma})$. The formulas for $\underline{\hat{\alpha}}^{(l+1)}$, $\underline{\hat{\theta}}^{(l+1)}$, and $\underline{\hat{\sigma}}^{(l+1)}$ are the same as before, but now the delays are estimated independently by maximizing the log likelihood directly with respect to one delay at a time:

$$\text{For } i = 1, \ldots, M$$

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} L_Z\left(\hat{\tau}_1^{(l+1)}, \ldots, \hat{\tau}_{i-1}^{(l+1)}, \tau_i, \hat{\tau}_{i+1}^{(l)}, \ldots, \hat{\tau}_M^{(l)}, \underline{\hat{\alpha}}^{(l)}, \underline{\hat{\theta}}^{(l)}, \underline{\hat{\sigma}}^{(l)}\right) \quad (112)$$

Substituting the formula for the likelihood function (13) and using straightforward algebraic manipulations gives:

ML-step:

$$\text{For } i = 1, \ldots, M$$

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \text{Re} \sum_{n=L}^{U} e^{-j\omega_n \tau_i} \hat{S}_i^{(l)}(\omega_n) Z_i^*(\omega_n) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)}) \quad (113)$$

where:

$$\hat{S}_i^{(l)}(\omega_n) = \frac{P_S(\omega_n; \underline{\hat{\theta}}^{(l)}) \left[\sum_{\substack{k=1 \\ k \neq i}}^{M} \hat{\alpha}_k^{(l)} e^{j\omega_n \hat{\tau}_k} Z_k(\omega_n) / P_{V_k}(\omega_n; \underline{\hat{\sigma}}_k^{(l)}) \right]}{1 + \sum_{k=1}^{M} \widehat{SNR}_k^{(l)}(\omega_n)} \quad (114)$$

where:

$$\hat{\tau}_k = \begin{cases} \hat{\tau}_k^{(l+1)} & \text{for } k < i \\ \hat{\tau}_k^{(l)} & \text{for } k > i \end{cases} \quad (115)$$

Equation (113) is similar in form to the EM algorithm estimate of $\hat{\tau}_i^{(l+1)}$ in (63). The new $\hat{\tau}_i^{(l+1)}$ estimate is obtained by maximizing a weighted cross-correlation between $Z_i^*(\omega_n)$ and and the signal estimate $\hat{S}_i^{(l)}(\omega_n)$. The difference between (63) and (113) is that the EM-ML algorithm uses a signal estimate formed from all but the i^{th} receiver

outputs. If we repeat the convergence rate analysis for the large WT case, we will find that each delay estimate approximately solves:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \sum_{\substack{k=1 \\ k \neq i}}^{M} \hat{R}_{s_{ik}}^{(l)} \left(\tau_i - \bar{\tau}_i + \bar{\tau}_k - \hat{\tau}_k \right) \tag{116}$$

This is an improvement over the behavior of the EM algorithm, since using $\hat{S}_i^{(l)}(\omega_n)$ rather than $\hat{S}^{(l)}(\omega_n)$ eliminates the extra peak in the vicinity of the old delay estimate. Initial convergence should therefore be improved, particularly in the case where the signal energy in all channels is equal.

With $M = 2$ channels, this algorithm is identical to the EM-ML joint delay estimation algorithm. We can therefore set $\tau_2 = 0$, optimize only over τ_1. and will achieve superlinear convergence. With $M > 2$, however, the convergence rate is no longer superlinear. Appendix B shows that when signal energy is equal in all channels, then the asymptotic convergence rate should be between $1/8$ and $1/9$, for any SNR above threshold and for any M. With unequal signal energy, we would expect faster convergence. Note that this convergence behavior is quite different than EM, where the rate depends on the SNR and on the number of channels. As with the EM algorithm, however, fastest convergence is achieved by estimating all M delays, and by estimating all M gains α_k, together with the signal spectral gain g^2.

This algorithm represents a compromise between the EM and the EM-ML joint delay algorithms. It requires slightly more effort than the EM algorithm, since we need to generate $\hat{S}_i^{(l)}(\omega_n)$ for $i = 1, \ldots, M$. However, we may expect significantly better convergence behavior. When compared with the EM-ML algorithm in (107), it replaces the $(M - 1)$ dimensional search for the delay parameters with a series of M searches for one delay at a time, thus trading off computational complexity for a slower rate of convergence. Each iteration increases the likelihood $L_Z(\underline{\xi})$, and convergence is guaranteed to a stationary point of the likelihood under mild conditions. The convergence rate, however, will only be linear.

6.3 EM-ML Delay/Gain Algorithm

As already indicated, the convergence of the delay estimates and their final error variance are somewhat insensitive to details of the signal and noise spectra. Therefore, if we are primarily interested in the delay estimation, it may be sufficient to iterate on the gain and spectral estimates a few times until they are "good enough", then iterate solely to improve the delay estimates.

However, in some applications, it may be important to find good estimates of the signal and noise parameters. For example, spectral shape and gain may be useful in identifying the target. Also, wildly varying signal gains may be an indicator of multipath distortion or sensor failure. In situations like these, it would be helpful to expend effort to accelerate the convergence of the gain and spectral parameters.

We can try a hybrid EM-ML algorithm which directly maximizes the log likelihood over $\underline{\tau}$, $\underline{\alpha}$, and $\underline{\sigma}$, and only uses an EM step to update the $\underline{\theta}$ estimates. Start with the same partitioning of the parameters as in the other EM-ML algorithms, then iteratively maximize $Q(\tilde{\underline{\xi}};\underline{\xi})$ with respect to $(\tilde{\underline{\xi}},\underline{\xi}_1)$, then $(\tilde{\underline{\xi}},\underline{\xi}_2)$, then $\underline{\xi}_3$. This hill climbing approach is similar to that of the EM-ML joint delay estimation algorithm, except that we maximize with respect to $\tilde{\underline{\xi}}$ twice per iteration. We would therefore expect this algorithm to converge more quickly to the ML solution. Following the reasoning in section 3.2, this algorithm can be shown to be equivalent to:

$$
\begin{aligned}
&\text{For } l = 0, 1, 2, \ldots \\
&\qquad \hat{\underline{\tau}}^{(l+1)} \leftarrow \max_{\underline{\tau}} L_Z\left(\underline{\tau}, \hat{\underline{\alpha}}^{(l)}, \hat{\underline{\sigma}}^{(l)}, \hat{\underline{\theta}}^{(l)}\right) \\
&\qquad \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\sigma}}^{(l+1)} \leftarrow \max_{\underline{\alpha},\underline{\sigma}} L_Z\left(\hat{\underline{\tau}}^{(l+1)}, \underline{\alpha}, \underline{\sigma}, \hat{\underline{\theta}}^{(l)}\right) \\
&\qquad \hat{\underline{\theta}}^{(l+1)} \leftarrow \max_{\underline{\theta}} \sum_{n=L}^{U} \left[\log P_S(\omega_n;\underline{\theta}) + \frac{\left|\widehat{S(\omega_n)}\right|^{2^{(l)}}}{P_S(\omega_n;\underline{\theta})}\right]
\end{aligned}
\tag{117}
$$

where $\left|\widehat{S(\omega_n)}\right|^{2^{(l)}}$ is computed using the latest parameter estimates $\hat{\underline{\tau}}^{(l+1)}$, $\hat{\underline{\alpha}}^{(l+1)}$, and $\hat{\underline{\sigma}}^{(l+1)}$. We could also maximize the likelihood over each delay variable one at a time. In either case, the likelihood will increase on each iteration, and convergence is guaranteed under mild conditions to a stationary point of the likelihood.

The delay estimation step is the same as in the EM-ML joint or individual delay algorithm, (107) or (113), and the spectral parameter estimate is the same as in the EM algorithm, (62). The new step is the calculation of $\hat{\underline{\alpha}}^{(l+1)}$ and $\hat{\underline{\sigma}}^{(l+1)}$. Unfortunately, in general the maximization required is highly nonlinear.

In certain limited circumstances, however, it is possible to find a closed form solution for the gain estimates. Let us assume that the noises on all receivers have the same spectra, with the same shape and the same unknown gain, $\sigma_i = \sigma$ for all i. Thus $P_{V_i}(\omega_n;\sigma) = \sigma P_V(\omega_n)$. Let us also assume that the parameters $\underline{\theta}$ include an overall signal gain level. This means that we can fix the average value of the α_i to an arbitrary level, and rely on $\underline{\theta}$ to control overall signal gain. One particularly useful normalization for the α_i is to restrict the gains to satisfy:

$$\frac{1}{\sigma}\sum_{i=1}^{M}\alpha_i^2 = \gamma \tag{118}$$

where γ is an arbitrary positive constant. Substituting (118) into (13), ignoring terms which do not depend on $\underline{\alpha}$ or σ, and recognizing that $\underline{\alpha}$ is restricted to be real:

$$L_Z(\underline{\xi}) = c - NM\log\sigma - \frac{1}{\sigma}\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)} + \frac{N}{\sigma^2}\underline{\alpha}^T\mathbf{R}(\underline{\tau},\underline{\theta})\underline{\alpha} \tag{119}$$

where $\mathbf{R}(\underline{\tau},\underline{\theta})$ is an $M\times M$ matrix of weighted receiver cross-correlations whose elements are:

$$\mathbf{R}_{ik}(\underline{\tau},\underline{\theta}) = \mathrm{Re}\frac{1}{N}\sum_{n=L}^{U}\frac{e^{j\omega_n(\tau_i-\tau_k)}Z_i(\omega_n)Z_k^*(\omega_n)P_S(\omega_n;\underline{\theta})/P_V^2(\omega_n)}{1+\gamma P_S(\omega_n;\underline{\theta})/P_V(\omega_n)} \tag{120}$$

To estimate $\hat{\underline{\alpha}}^{(l+1)}$ and $\hat{\sigma}^{(l+1)}$, we need to maximize (119) subject to the constraint $\underline{\alpha}^T\underline{\alpha} = \gamma\sigma$. Using the method of Lagrange multipliers, the solution to this constrained maximization is derived in Appendix A. First we compute the largest eigenvalue $\mu^{(l)}$ of $\mathbf{R}_{ik}(\hat{\underline{\tau}}^{(l+1)},\hat{\underline{\theta}}^{(l)})$, together with the corresponding eigenvector $\underline{v}^{(l)}$. (There are numerous computationally efficient methods for extracting the largest eigenvalue and eigenvector.) Then:

ML-step:

$$\hat{\sigma}^{(l+1)} \leftarrow \frac{1}{NM}\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)} - \mu^{(l)}\frac{\gamma}{M} \tag{121}$$

$$\hat{\underline{\alpha}}^{(l+1)} = \left(\frac{\gamma\hat{\sigma}^{(l+1)}}{\underline{v}^T\underline{v}}\right)^{1/2}\underline{v}^{(l)} \tag{122}$$

To analyze the convergence rate, assume that the observation interval is sufficiently large, $WT/2\pi \gg 1$, so that the matrix $\mathbf{R}(\hat{\underline{\tau}}^{(l+1)}, \hat{\underline{\theta}}^{(l)})$ is close to its expected value conditioned on the correct (but unknown) parameters. Let us further assume that the delays are known exactly, $\hat{\underline{\tau}}^{(l+1)} = \bar{\underline{\tau}}$. In this case, using (10):

$$\mathrm{E}_{\underline{\xi}}\left[\mathbf{R}(\bar{\underline{\tau}}, \hat{\underline{\theta}}^{(l)})\right] = \phi_s(\hat{\underline{\theta}}^{(l)})\bar{\underline{\alpha}}\bar{\underline{\alpha}}^T + \phi_v(\hat{\underline{\theta}}^{(l)})\mathbf{I} \tag{123}$$

where:

$$\phi_s(\hat{\underline{\theta}}^{(l)}) = \frac{1}{N}\sum_{n=L}^{U} \frac{P_S(\omega_n;\hat{\underline{\theta}}^{(l)})P_S(\omega_n;\bar{\underline{\theta}})/P_V^2(\omega_n)}{1+\gamma P_S(\omega_n;\hat{\underline{\theta}}^{(l)})/P_V(\omega_n)} \tag{124}$$

$$\phi_v(\hat{\underline{\theta}}^{(l)}) = \frac{\bar{\sigma}}{N}\sum_{n=L}^{U} \frac{P_S(\omega_n;\hat{\underline{\theta}}^{(l)})/P_V(\omega_n)}{1+\gamma P_S(\omega_n;\hat{\underline{\theta}}^{(l)})/P_V(\omega_n)} \tag{125}$$

where we assume that $\bar{\underline{\alpha}}$, the correct gain, satisfies the constraints, $\bar{\underline{\alpha}}^T\bar{\underline{\alpha}}/\gamma\bar{\sigma} = 1$. The largest eigenvalue of the matrix in (123) is:

$$\mu^{(l)} = \gamma\bar{\sigma}\phi_s(\hat{\underline{\theta}}^{(l)}) + \phi_v(\hat{\underline{\theta}}^{(l)}) \tag{126}$$

and the corresponding eigenvector is:

$$\underline{v}^{(l)} = \bar{\underline{\alpha}} \tag{127}$$

Thus under the large WT assumption with known delays, the estimate of the gains converges in one step (superlinearly) to the correct values, $\hat{\underline{\alpha}}^{(l+1)} = \bar{\underline{\alpha}}$. The noise gain estimate, however, depends on the eigenvalue $\hat{\mu}^{(l)}$, which in turn depends on the latest estimate of the signal spectrum parameters, $\hat{\underline{\theta}}^{(l)}$. Assuming large $WT/2\pi \gg 1$:

$$\begin{aligned}
\hat{\sigma}^{(l+1)} &= \frac{1}{MN}\sum_{n=L}^{U}\sum_{i=1}^{M}\mathrm{E}_{\underline{\xi}}\left[\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)}\right] - \hat{\mu}^{(l)}\frac{\gamma}{M} \\
&= \frac{1}{MN}\sum_{n=L}^{U}\sum_{i=1}^{M}\left[\frac{\bar{\alpha}_i^2 P_S(\omega_n;\bar{\underline{\theta}}) + \bar{\sigma}P_V(\omega_n)}{P_V(\omega_n)}\right] - \frac{\gamma^2\bar{\sigma}\phi_s(\hat{\underline{\theta}}^{(l)}) + \gamma\phi_v(\hat{\underline{\theta}}^{(l)})}{M} \\
&= \bar{\sigma}\left[1 + \frac{\gamma}{NM}\sum_{n=L}^{U}\frac{\left[P_S(\omega_n;\bar{\underline{\theta}}) - P_S(\omega_n;\hat{\underline{\theta}}^{(l)})\right]/P_V(\omega_n)}{1+\gamma P_S(\omega_n;\hat{\underline{\theta}}^{(l)})/P_V(\omega_n)}\right]
\end{aligned} \tag{128}$$

Since $P_S(\omega_n;\hat{\underline{\theta}}^{(l)})$ converges at a linear (though rapid) rate to $P_S(\omega_n;\bar{\underline{\theta}})$, $\hat{\underline{\sigma}}^{(l+1)}$ will converge at a linear (and rapid) rate to $\bar{\sigma}$.

6.4 ML Coordinate Ascent Algorithm

As the final step, we may try to replace the EM-like estimation of $\hat{\underline{\theta}}^{(l+1)}$ with a direct maximization of the original log likelihood function. With the same partitioning of the parameters as before, maximize $Q(\tilde{\underline{\xi}};\underline{\xi})$ with respect to $(\tilde{\underline{\xi}},\underline{\xi}_1)$, then $(\tilde{\underline{\xi}},\underline{\xi}_2)$, then $(\tilde{\underline{\xi}},\underline{\xi}_3)$, and then iterate. Because this algorithm maximizes over $\tilde{\underline{\xi}}$ three times per iteration, we would expect it to converge more quickly than any of the other algorithms.

The delay, gain, and noise parameter estimates are found by maximizing the likelihood function as in the previous algorithm. The spectral parameters, however, are now also calculated by direct maximization of the likelihood function:

$$\hat{\underline{\theta}}^{(l+1)} \leftarrow \max_{\underline{\theta}} L_Z\left(\hat{\underline{\tau}}^{(l+1)}, \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\sigma}}^{(l+1)}, \underline{\theta}\right) \tag{129}$$

The complete algorithm is now just a direct coordinate ascent search over the observed data likelihood $L_Z(\underline{\xi})$.

In general, this optimization is quite complicated, except in the special case that we model the signal spectrum as a non-parametric unknown function $P_S(\omega_n)$. Let us also restrict the noise spectra in the same way as in the EM-ML delay/gain estimation algorithm, so that all noise spectra are identical, with the same unknown gain, $\sigma_i = \sigma$ and with $P_{V_i}(\omega_n;\sigma) = \sigma P_V(\omega_n)$. Also constrain the signal gains with formula (118). Then, following straightforward but tedious algebra, we obtain the following closed form solution to (129):

$$\hat{P}_S^{(l+1)}(\omega_n) = \frac{\left|\sum_{i=1}^{M} \hat{\alpha}_i^{(l+1)} e^{j\omega_n \hat{\tau}_i^{(l+1)}} Z_i(\omega_n)\right|^2}{\gamma^2 \hat{\sigma}^{(l+1)^2}} - \frac{P_V(\omega_n)}{\gamma} \tag{130}$$

For frequency samples at which this expression is below some small positive value ϵ, the signal spectrum estimate should be set to ϵ.

To assess the convergence rate of this procedure, we assume WT is large, so that the expression above tends to its expected value based on the correct (but unknown) parameters. Assume that the delay estimates are correct, $\hat{\underline{\tau}}^{(l+1)} = \bar{\underline{\tau}}$. Using (13) and following straightforward algebra:

$$\mathrm{E}_{\underline{\xi}}\left[\hat{P}_S^{(l+1)}(\omega_n)\right] = \bar{P}_S(\omega_n)\left(\frac{1}{\gamma\hat{\sigma}^{(l)}}\sum_{i=1}^{M}\hat{\alpha}_i^{(l)}\bar{\alpha}_i\right)^2 + \frac{1}{\gamma}P_V(\omega_n)\left(\frac{\bar{\sigma}}{\hat{\sigma}^{(l)}} - 1\right) \tag{131}$$

If the correct gains and noise spectral parameters were known, $\hat{\underline{\alpha}}^{(l)} = \underline{\bar{\alpha}}$ and $\hat{\underline{\sigma}}^{(l)} = \underline{\bar{\sigma}}$, then we would have $\mathrm{E}_{\underline{\xi}}\left[\hat{P}_S^{(l+1)}(\omega_n)\right] = \bar{P}_S(\omega_n)$, and convergence would be superlinear. Otherwise, the signal spectral estimate will converge at about the same rate that $\hat{\alpha}_i^{(l)}$ converges to $\bar{\alpha}_i$ and $\hat{\sigma}^{(l)}$ converges to $\bar{\sigma}$.

6.5 Comparison of the EM-ML Algorithms

Note that the EM-ML algorithms represent an interesting tradeoff between computational complexity and convergence rate. The EM algorithm in section 3 represents the computationally simplest approach. Its convergence rate, however, is only linear. If we replace the delay estimation step with the more complicated $(M-1)$ dimensional search of the EM-ML problems, then the convergence rate of the delays becomes super-linear, although the other parameters converge linearly. If we restrict the form of the noise power spectrum, and apply the gain and noise level EM-ML estimation algorithm, then we need to solve a maximum eigenvalue and eigenvector problem, but the convergence rate of the gains becomes super-linear. Finally, if we use a non-parametric signal spectral model, and apply the EM-ML signal spectrum algorithm, then we achieve super-linear convergence of the signal spectrum. Computationally simpler EM-ML delay estimation algorithms can also be derived by maximizing the likelihood with respect to one delay at a time. The disadvantage is that the convergence rates will only be linear.

7 Simulation Results

In this section we present some simulation results to illustrate the behavior of the EM and EM-ML algorithms. In general, the results of the simulation follow the predictions of the theoretical analysis we have given.

7.1 Simulation Details

We start with Gaussian signals with unit spacing between samples, generated by passing white Gaussian noise through an ARMA (pole-zero) filter. In the first set of examples we present, we used a 3 pole, low pass filter with a strong resonance at around .123 Hz. The signal power spectrum is:

$$P_S(\omega_n;\underline{\bar{\theta}}) = \frac{1}{\left|(1-0.9e^{-j\omega_n})(1-1.92\cos(.246\pi)e^{-j\omega_n}+0.9216e^{-j2\omega_n})\right|^2} \tag{132}$$

Figure 2a shows the signal power spectrum, and figure 2b shows the signal correlation. The noises on all receivers are white Gaussian, and have the same power, $\bar{\sigma}$, on all channels:

$$P_{V_i}(\omega_n;\bar{\sigma}) = \bar{\sigma} \tag{133}$$

To avoid having to use special treatment at the boundaries of each frame of data, and to avoid dealing with windowing issues, we synthesized all signals and noises $S(\omega_n)$ and $V_i(\omega_n)$ directly in the frequency domain, with sample spacing $2\pi/N$. The DC samples $S(0)$ and $V_i(0)$ were real Gaussian zero mean variables with variance $P_S(0;\underline{\bar{\theta}})$ and $\bar{\sigma}$ respectively. All other samples $S(\omega_n)$ and $V_i(\omega_n)$ were complex Gaussian zero mean variables with variance $P_S(\omega_n;\underline{\bar{\theta}})$ and $\bar{\sigma}$ respectively. Samples $S(\pi)$ and $V_i(\pi)$ were set to zero. This method generates signals and noises having periodic correlation functions, with period N. For these special sequences, the frequency domain formulas for the likelihoods which are derived in this paper are exactly correct.

We also synthesized the receiver outputs $Z_i(\omega_n)$ in the frequency domain, phase shifting the signal $S(\omega_n)$ as in (5) in order to achieve time shifts which are not limited to integer multiples of the sample period.

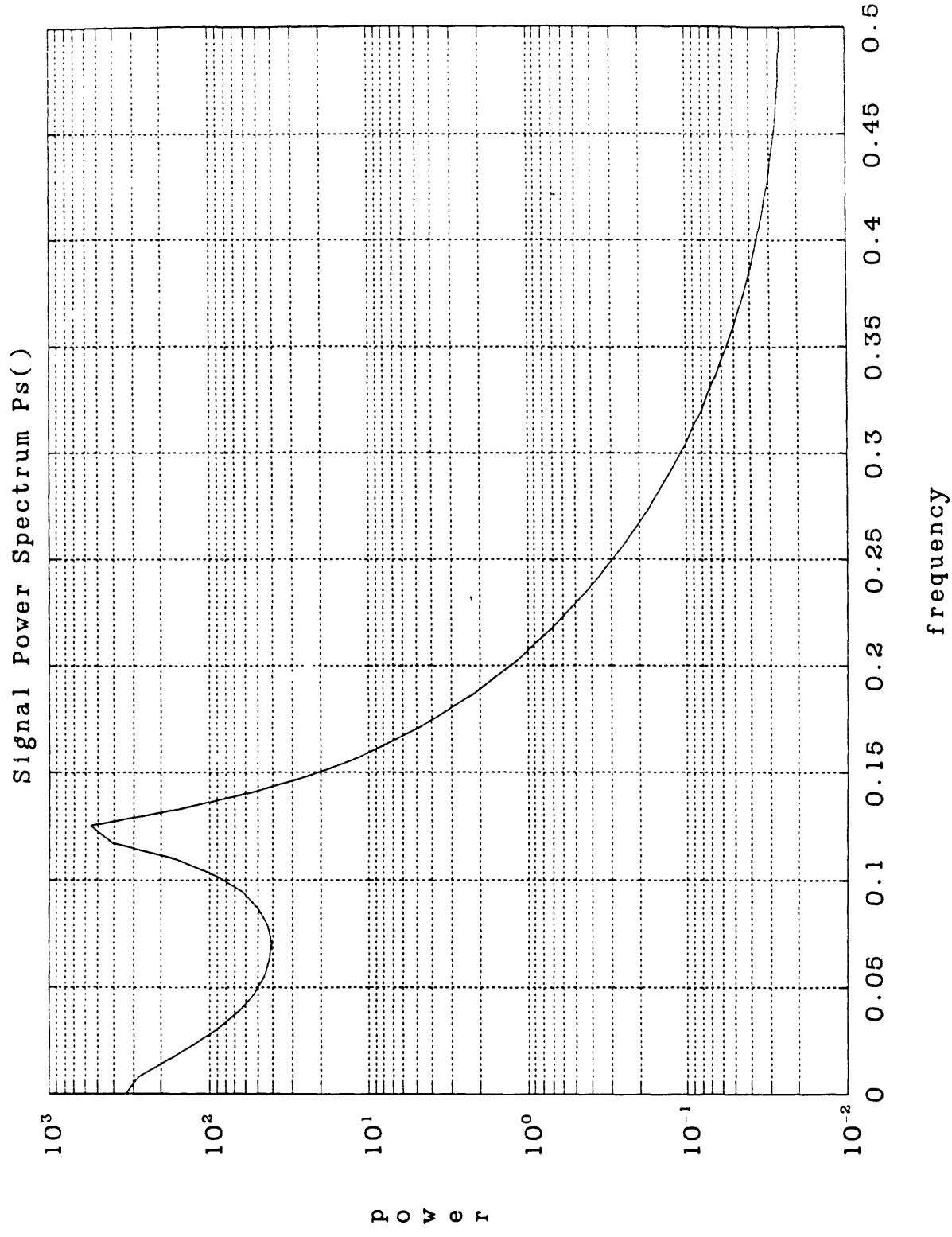


Figure 2a - Ideal 3-pole Lowpass Signal Power Spectrum

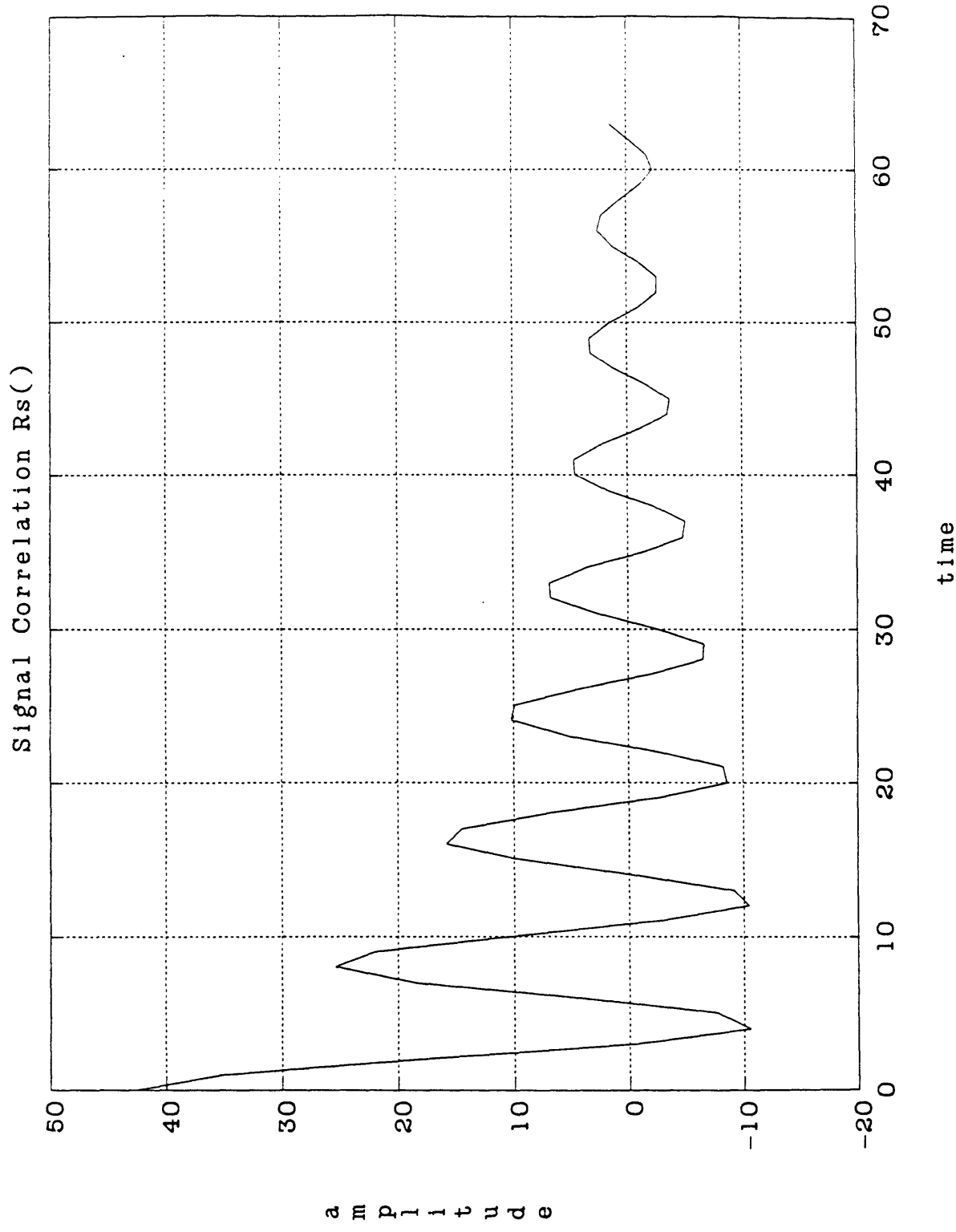


Figure 2b - Ideal 3-pole Lowpass Signal Correlation

The noise levels $\bar{\sigma}$ were set to meet a specified array signal-to-noise ratio (SNR), where we define:

$$SNR = 10\log_{10}\left(\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{\bar{\alpha}_i^2}{\bar{\sigma}}P_S(\omega_n;\bar{\underline{\theta}})\right) \tag{134}$$

To use all the available data, we modified the formulas for the likelihood functions in order to include the DC term. Because this term is real while the others are complex, the DC contribution to the likelihoods must be multiplied by $1/2$. We achieve the same effect by setting $L = -(N/2-1)$ and $U = N/2-1$, thus summing over both positive and negative frequencies when forming the likelihoods.

Another minor adjustment that is necessary in practice is that most FFT subroutines do not normalize the transform as assumed in (3). Thus FFT subroutine outputs must be normalized by $1/\sqrt{N}$ when computing the likelihood formulas.

The most difficult step in this algorithm is solving the maximizations required for the delays. We first do a coarse search, using a rather conservative procedure to improve the accuracy. Form a 4N point transform $F_i(\omega_n)$ as follows:

$$F_i\left(\frac{2\pi n}{4N}\right) = \begin{cases} \hat{S}^{(l)*}\left(\frac{2\pi n}{N}\right)Z_i\left(\frac{2\pi n}{N}\right) & n = 0,\ldots,N/2-1 \text{ and } n = 4N-N/2+1,\ldots,4N-1 \\ 0 & \text{else} \end{cases} \tag{135}$$

Take the real part of the inverse Fourier Transform and find the largest sample. This is the initial coarse estimate of the time delay. It should be within $1/4$ sample of the peak delay value. Note that oversampling is used in the coarse search step to minimize difficulties caused when the global peak of the cross-correlation lies between time samples. Without oversampling, we might underestimate the height of the global peak, and therefore place the delay estimate at a lesser peak which might happen to be centered at a sample location.

The coarse delay estimate is then refined by using binary search to locate the peak of the inverse DFT of $F_i(\omega)$ to within the minimum of .001 sample or .001 times the square root of the Cramer-Rao lower bound (the expected standard deviation of the delay estimate).

7.2 Simulation of the EM Algorithm

Consider a two-channel problem, $M = 2$, with $N = 128$ points, and with delays $\underline{\tau} = (-1.58 \;\; 0)^T$ and equal signal gains $\underline{\alpha} = (3 \;\; 3)^T$. Each step of the EM algorithm estimates the signal and its variance (in the frequency domain), then maximizes the cross-correlation between the signal estimate and each receiver signal (63) to estimate each delay. The gains are found from the normalized peak of this cross-correlation function. The signal power spectrum is estimated by fitting an all-pole model to the expected value of the signal periodogram, and the noise levels are individually estimated from the error between each receiver output and the appropriately scaled and phase shifted signal estimate.

Our first simulation estimates two delays and two gains, but sets the signal and noise power spectra to their correct values. Figures 3 show the estimates generated under four different array SNR: -10dB, 0dB, 10dB, and 30dB. For each case, we ran 20 simulations with different data, and iterated on each for 10 steps. In this and all subsequent examples, the initial delay estimates were set to 0, and the initial gains were set to 1. We show the behavior of the delay estimates, and the gain estimates, showing the correct values with dotted lines. We also plot the standard deviation of the final relative delay estimates from the correct value of -1.58 against the Cramer-Rao lower bound, given in (18). For comparison purposes, we also plot the standard deviation of the generalized cross correlation method (GCC) given by (23), where all the gains and spectra are known, and against an unweighted cross-correlation method (CC) with the window function set to $W_{12}(\omega_n) = 1$, run on the same data. Note the strong thresholding effect. Below a certain SNR, near 0dB, all methods sometimes confuse a noise peak at a random delay between -64 and +64 with the correct signal peak. This causes a precipitous increase in the measured standard deviation of the estimates. (The EM method in this case appears to have a better threshold than both GCC and CC, but this is a statistical fluke of this particular example, and does not occur in other runs.) Above the threshold, the methods choose the correct peak, but are able to locate it with differing degrees of accuracy. Careful examination of the delay estimates

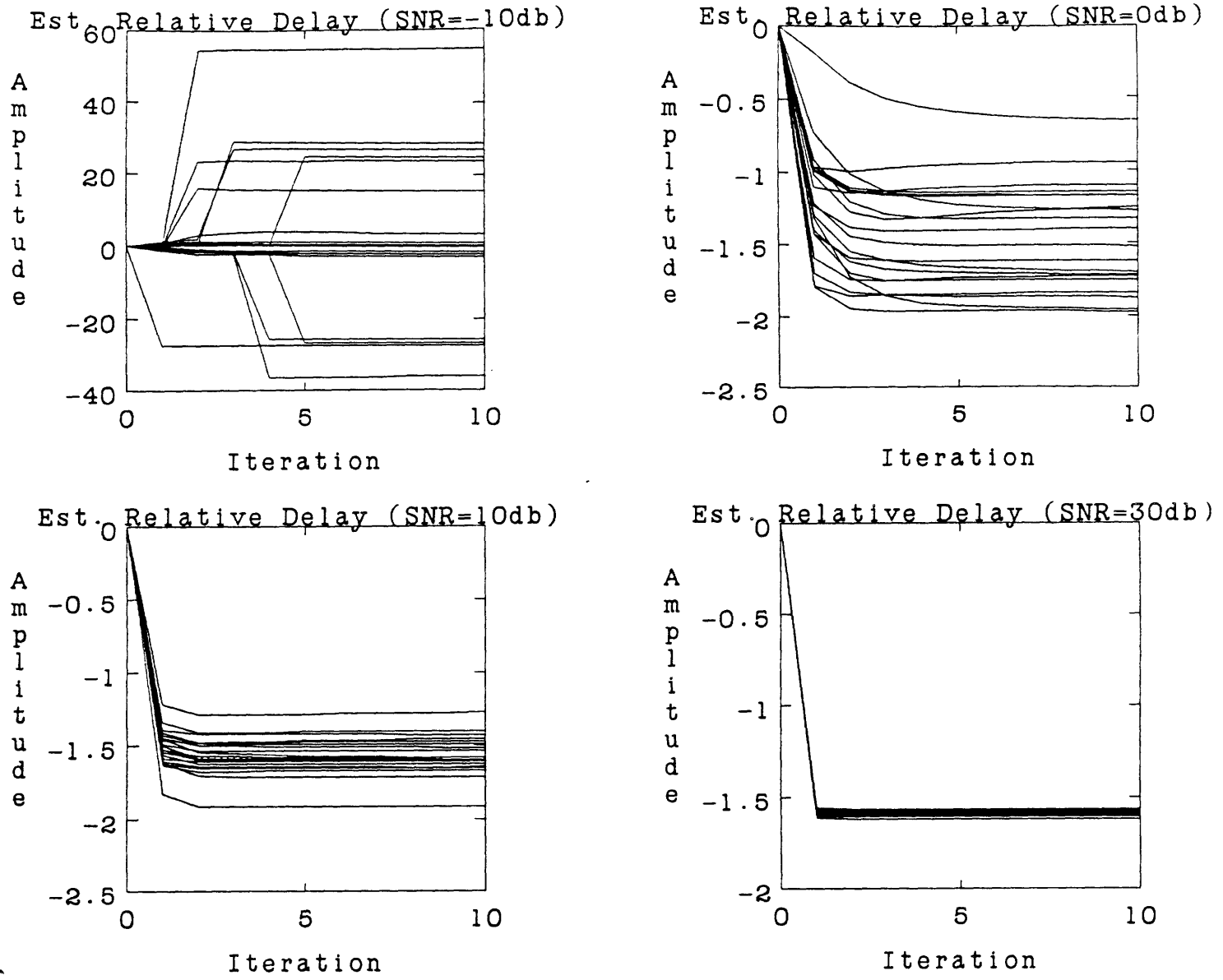


Figure 3a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known lowpass signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains.

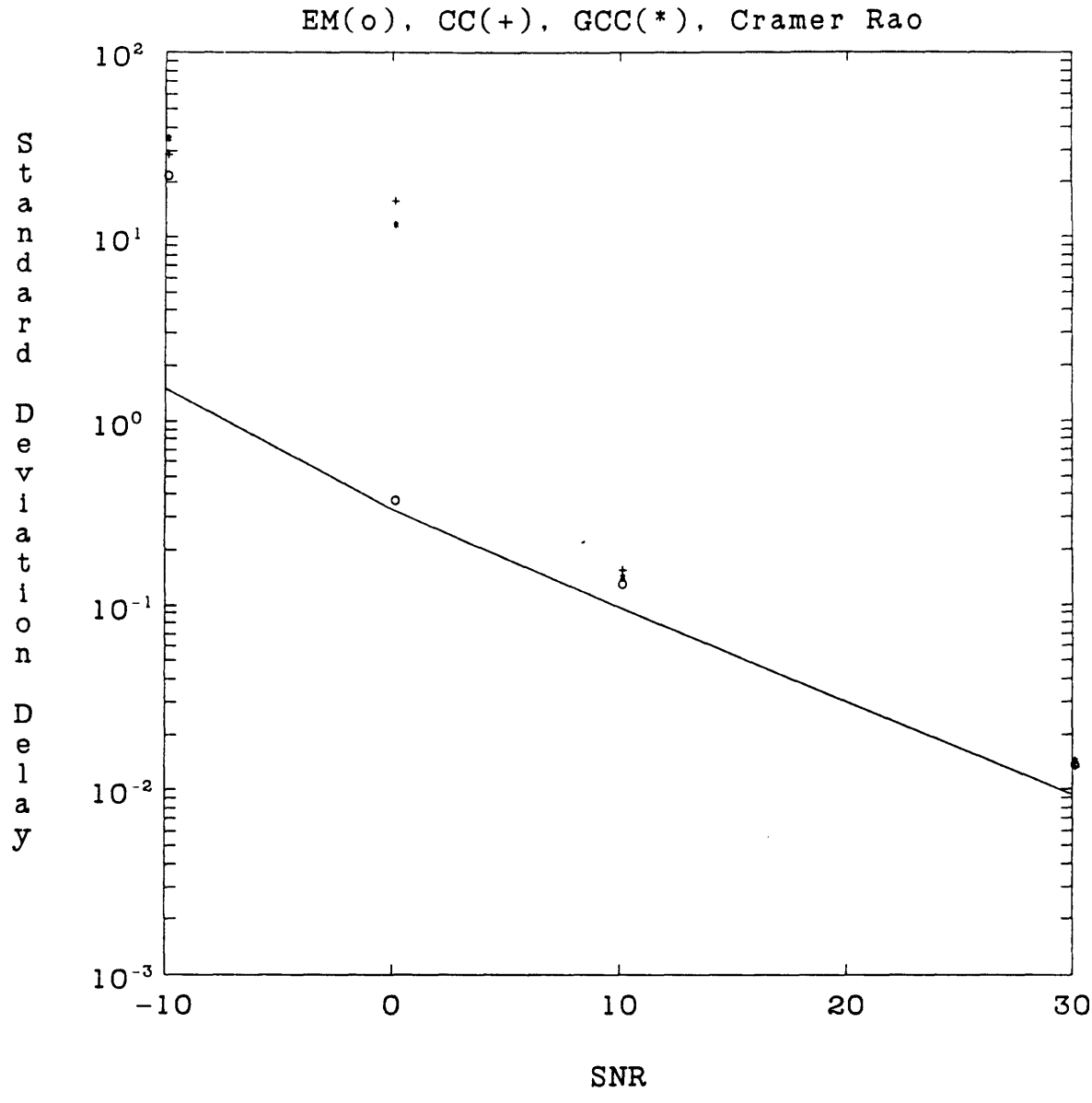


Figure 3b - EM Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

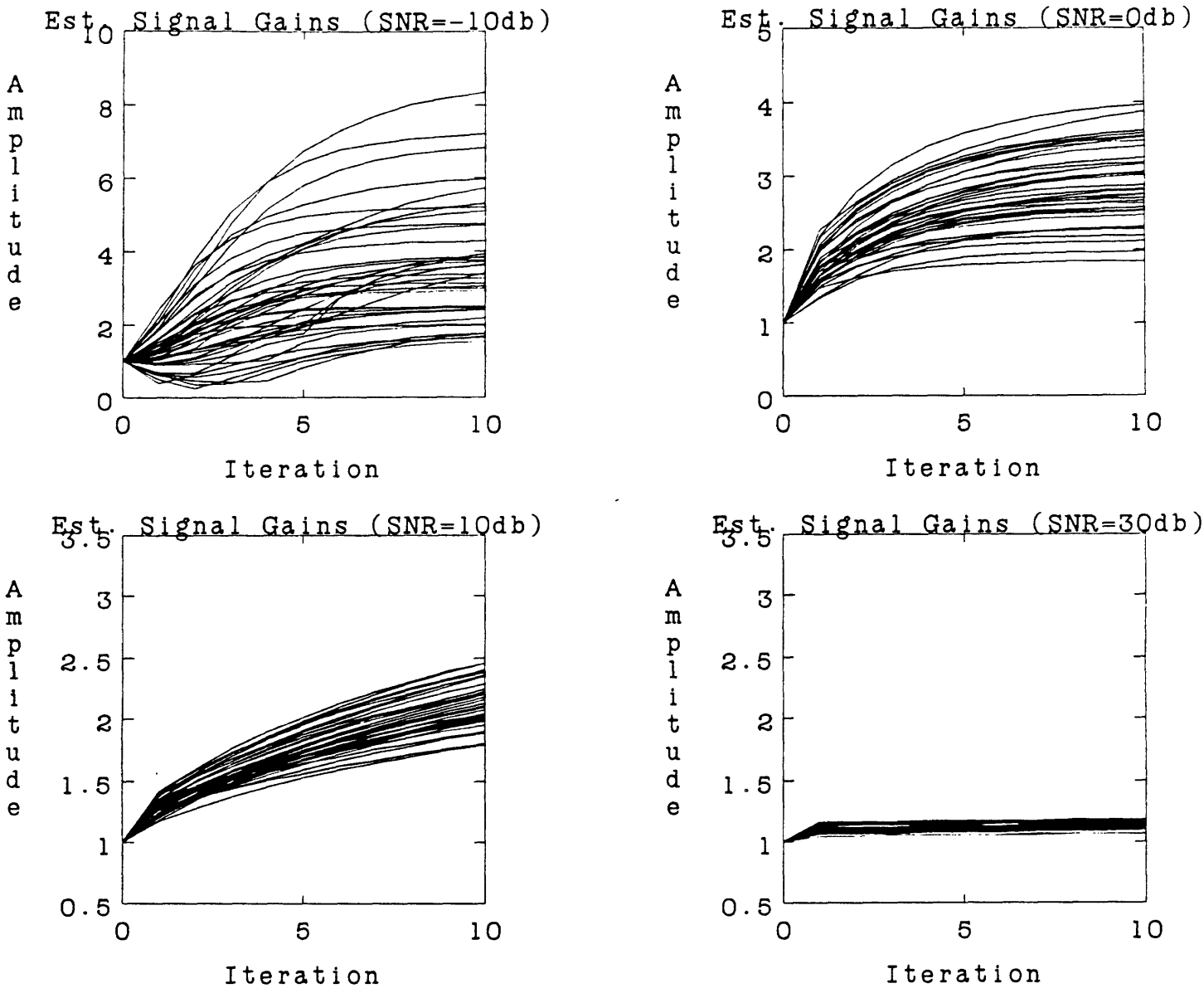


Figure 3c - EM Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$

in general shows that the EM delay estimates have variance which is similar to that of GCC, and significantly better than CC. Note also the relatively rapid convergence of the EM delay estimates. As predicted, convergence is fastest at high SNR (about 2 iterations). The EM gain estimates, on the other hand, converge much more slowly, particularly at high SNR. This is consistent with our previous analysis which suggested fast convergence of the relative values of the gains, but slow convergence of their average level.

Figures 4 illustrate the behavior of the EM algorithm when we set $\tau_2 = 0$, and only estimate τ_1. We also estimate both gains α_1, α_2. From the plot of the standard deviation of the final delay estimates, it is clear that we achieve the same ultimate accuracy. However, the convergence rate of the delays is more than twice as slow as when we estimate both delays on every iteration.

Next, figures 5 illustrate the behavior of the EM algorithm when we estimate *all* the parameters. On each iteration, we estimate both delays, both signal gains, a 6 pole signal power spectrum with arbitrary gain g^2 (the original data was only a 3 pole spectrum), and every noise gain (the original data had all noise levels equal). The initial signal power spectrum estimate is flat, with energy equal to the average energy in the received data. All initial noise level estimates were set to 1. Despite the fact that we use only estimated spectra when estimating the delays, for SNR above the threshold the standard deviation of the final delay estimates is virtually unchanged, and still approaches that of the GCC algorithm. The convergence rate of the delays, however, is slower than in figure 3 where the spectra were known. As predicted, the convergence rate depends strongly on the SNR, with fastest convergence (about 2 iterations) at 30dB. Note that the signal gains do not converge to their correct values. This is because we have deliberately introduced an ambiguity in the signal gain level, allowing both the gains $\underline{\alpha}$ and the spectral gain g^2 to control the effective signal level. Only the product $g\ \underline{\alpha}$ can be identified uniquely from the data. Convergence of the signal power spectrum estimate, however, is quite a bit faster than that of the signal gains. The result of introducing this gain ambiguity is that the signal spectrum level shifts up to capture the average signal level in all channels, while the gains $\hat{\underline{\alpha}}^{(l)}$ adapt

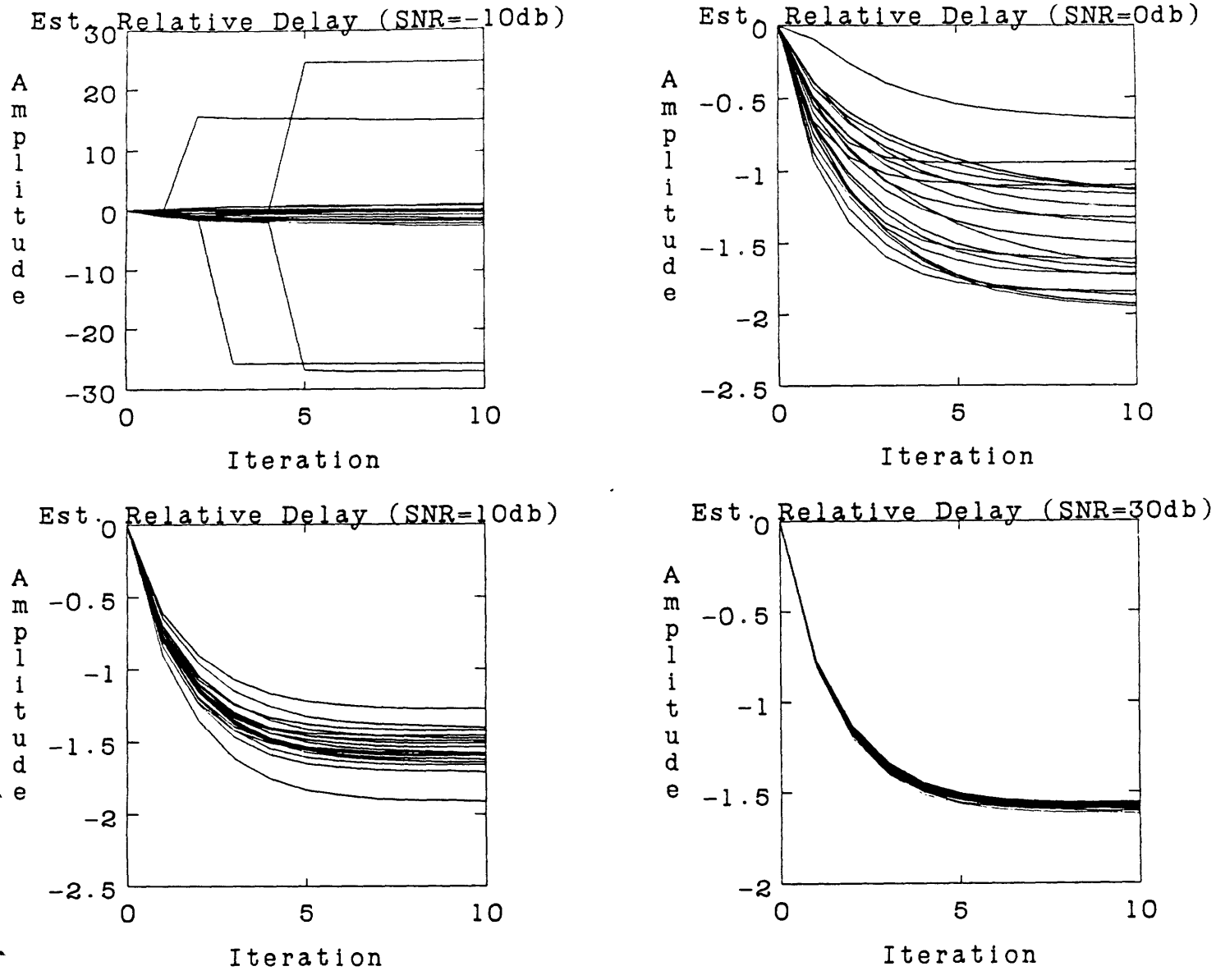


Figure 4a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known lowpass signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (33)^T$, 20 runs for each SNR, estimate 1 delay, 2 gains.

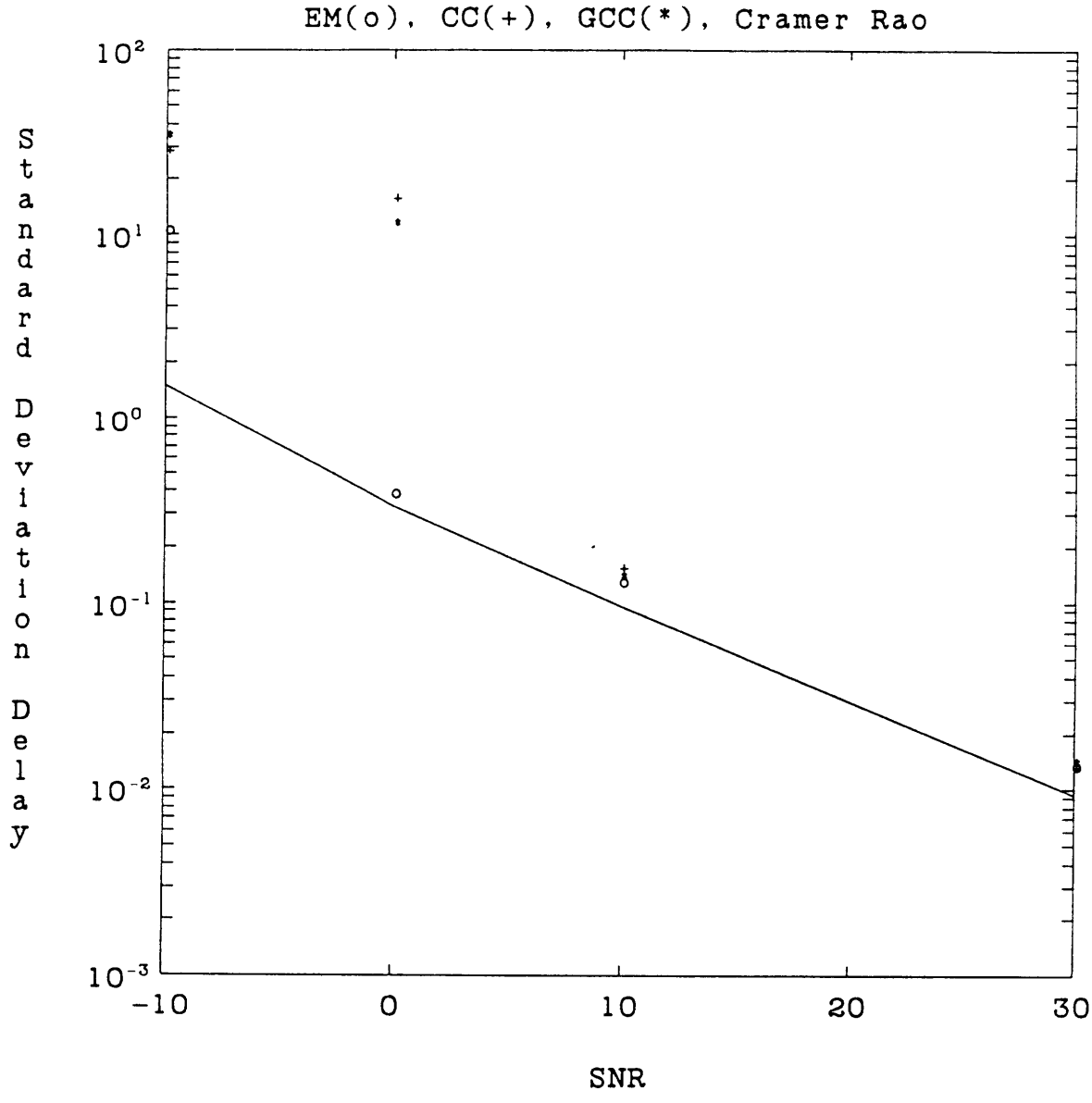


Figure 4b - EM Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

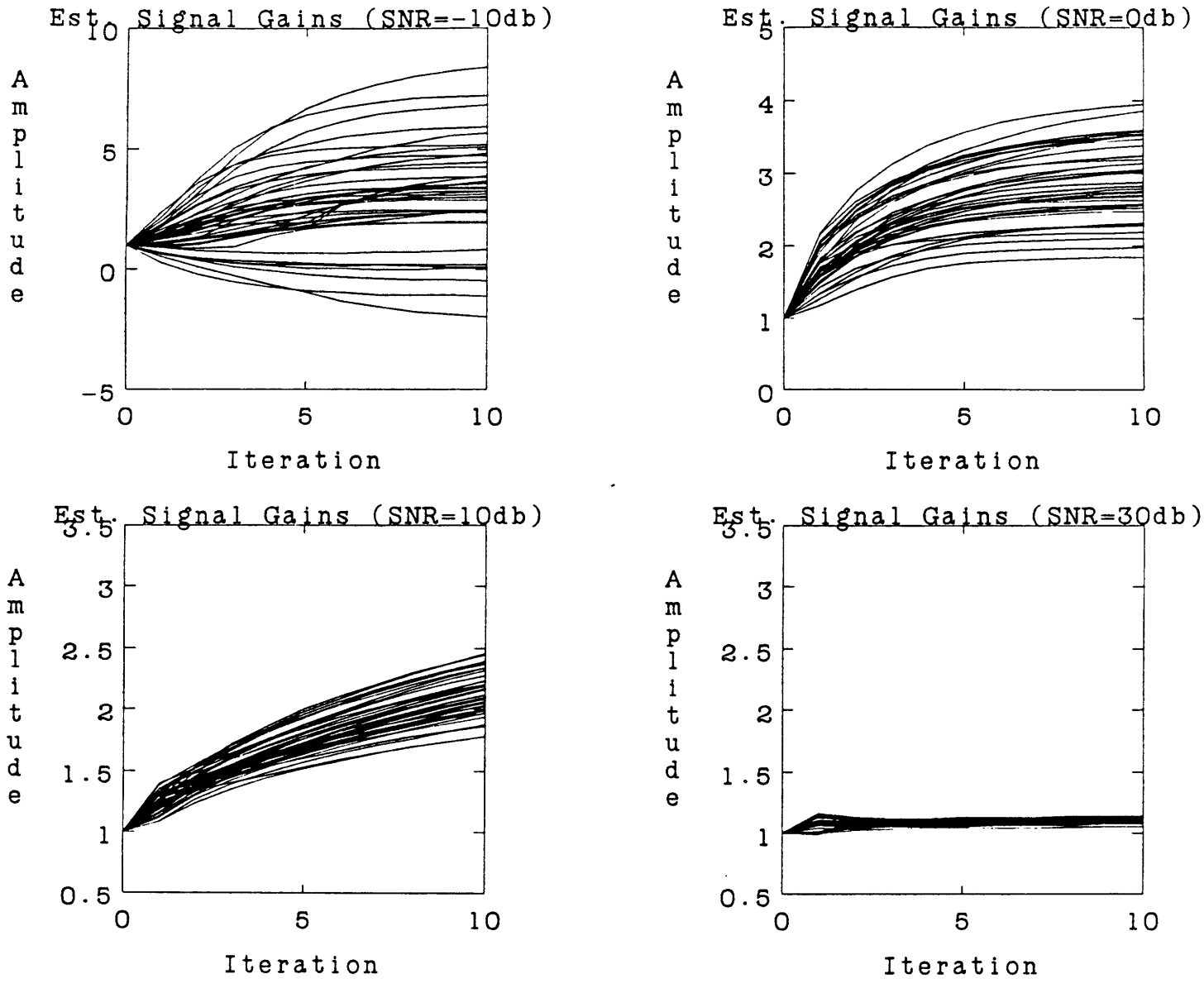


Figure 4c - EM Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$.

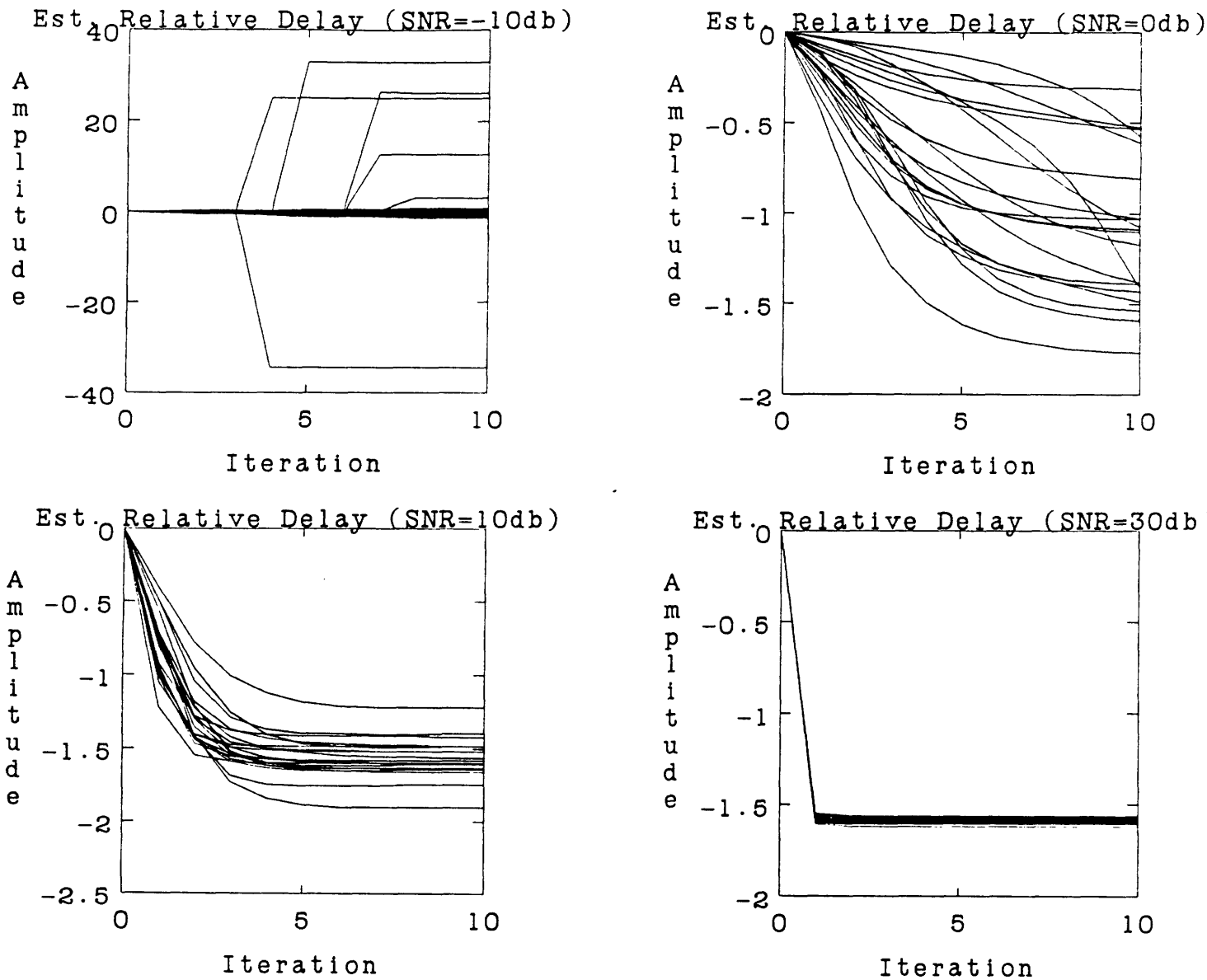


Figure 5a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains, 6 pole signal power spectrum with gain, 2 noise level parameters.

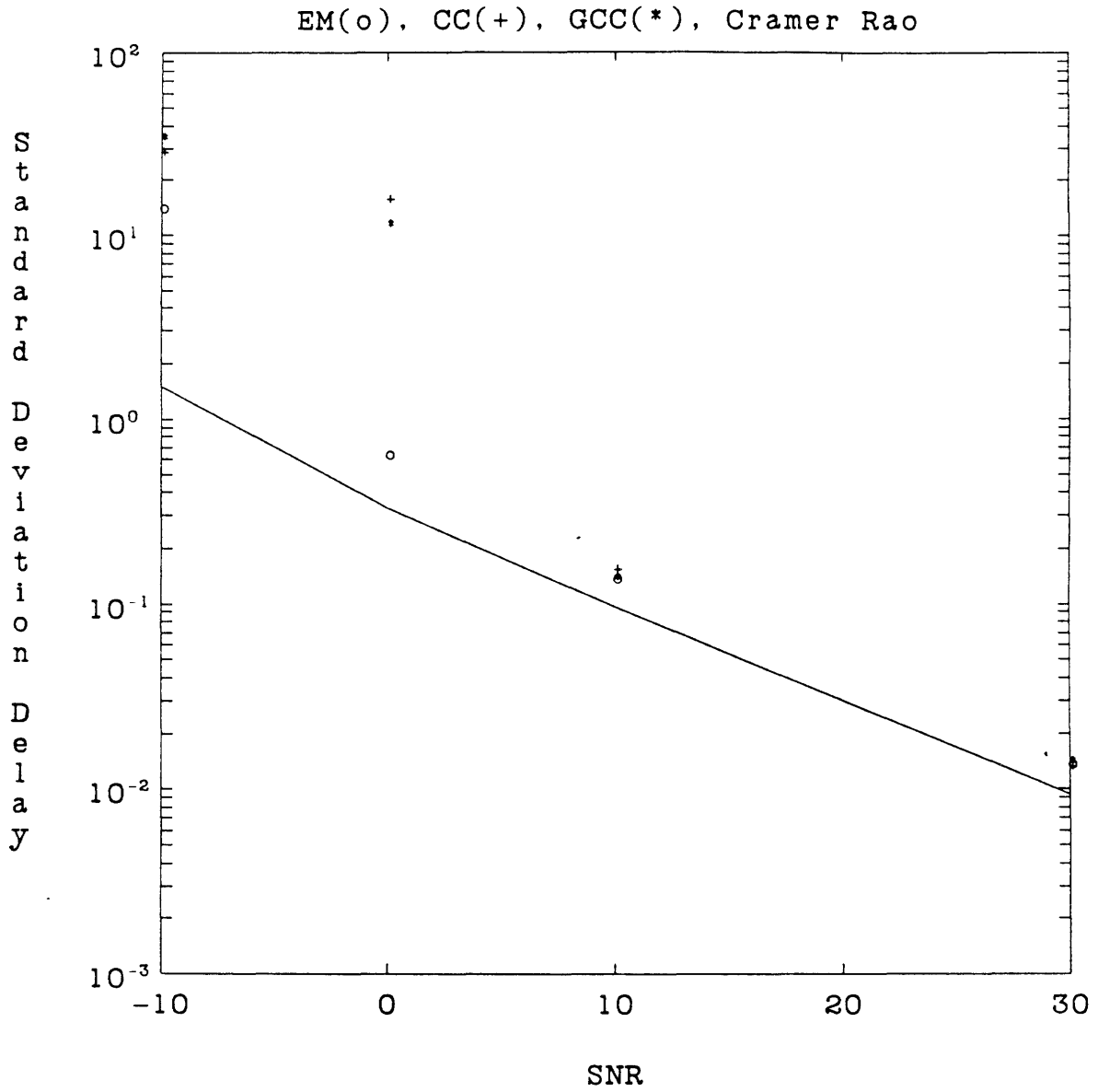


Figure 5b - EM Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

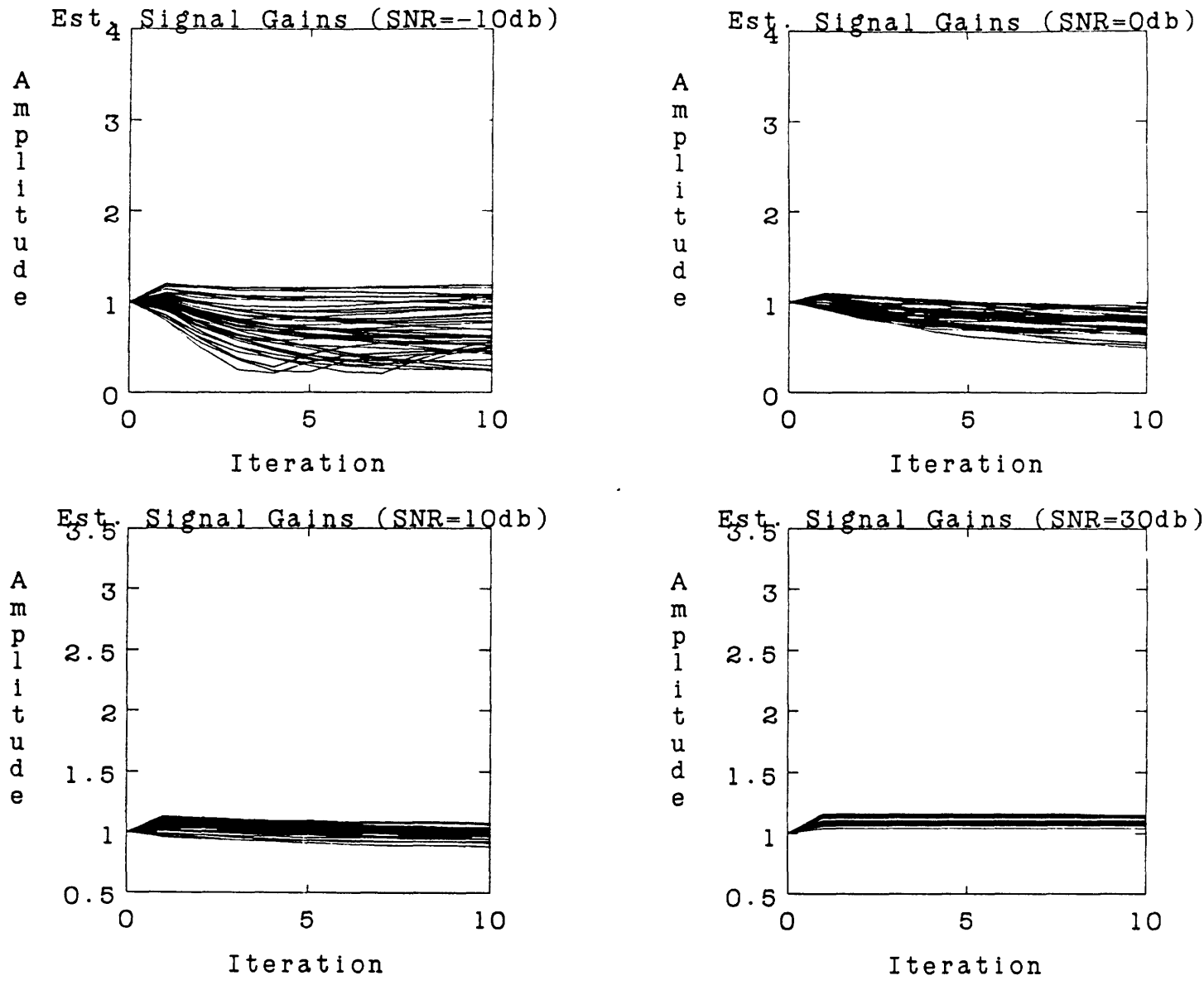


Figure 5c - EM Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$.

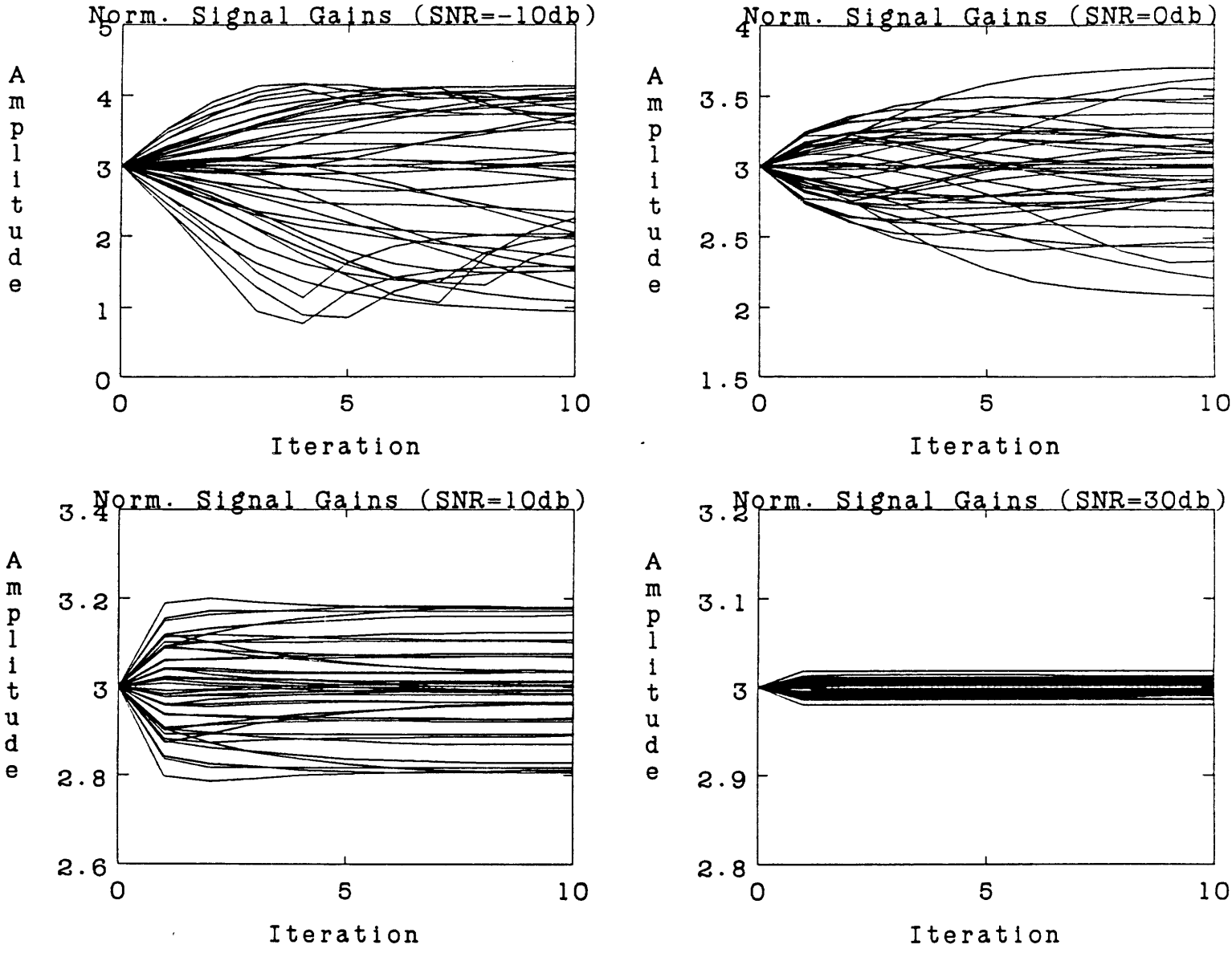


Figure 5d - EM Algorithm, Normalized Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

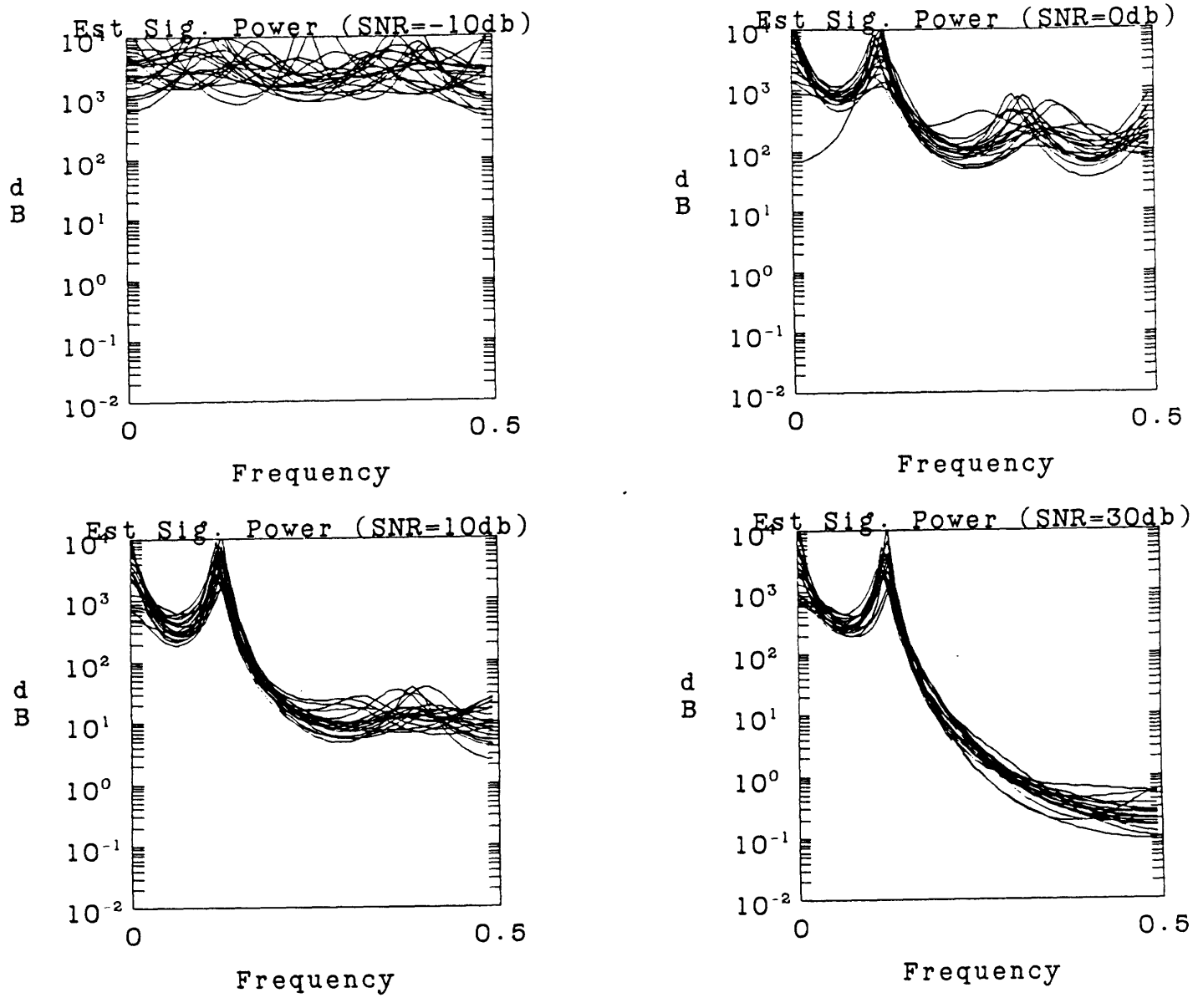


Figure 5e - EM Algorithm, Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)})$.

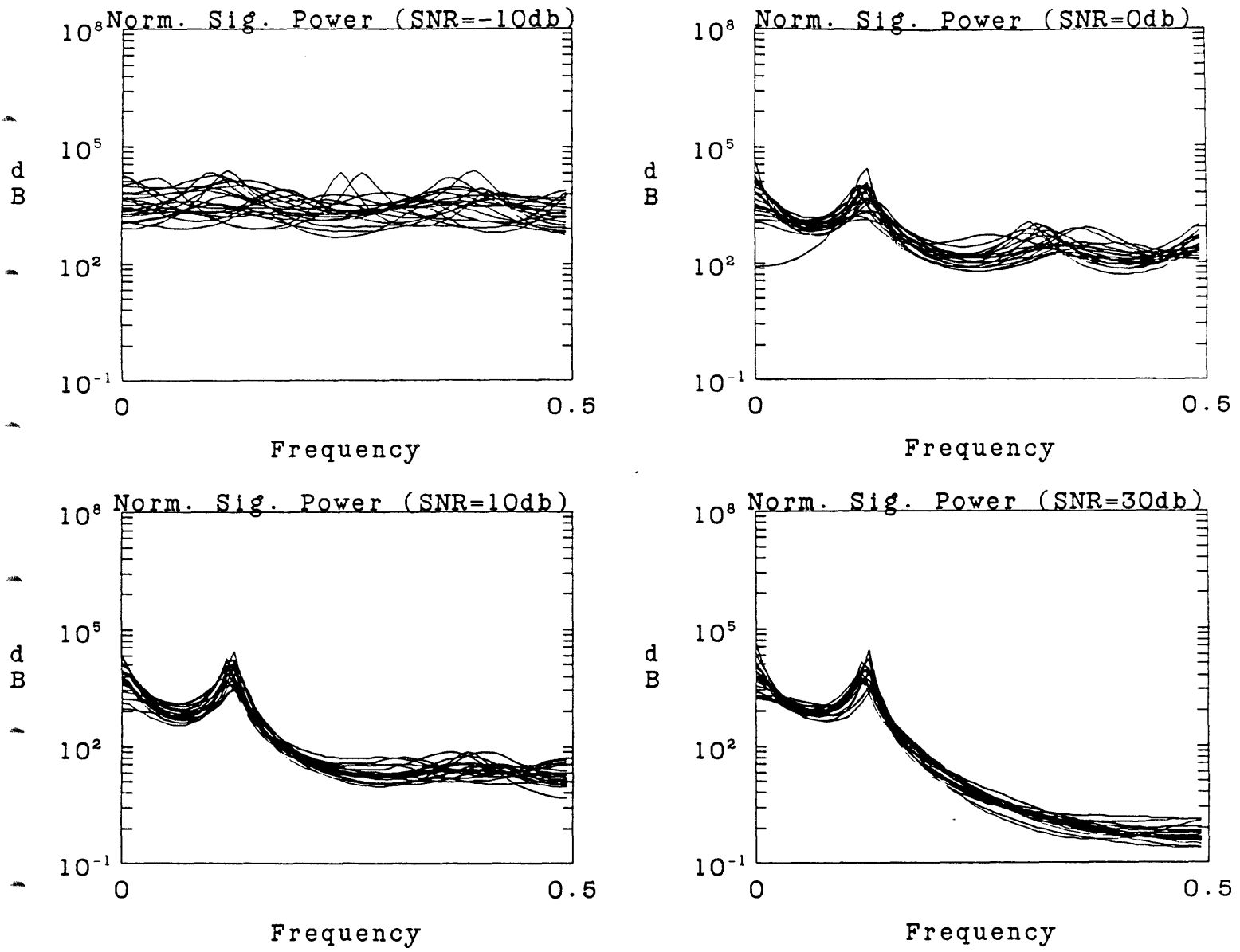


Figure 5f - EM Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$

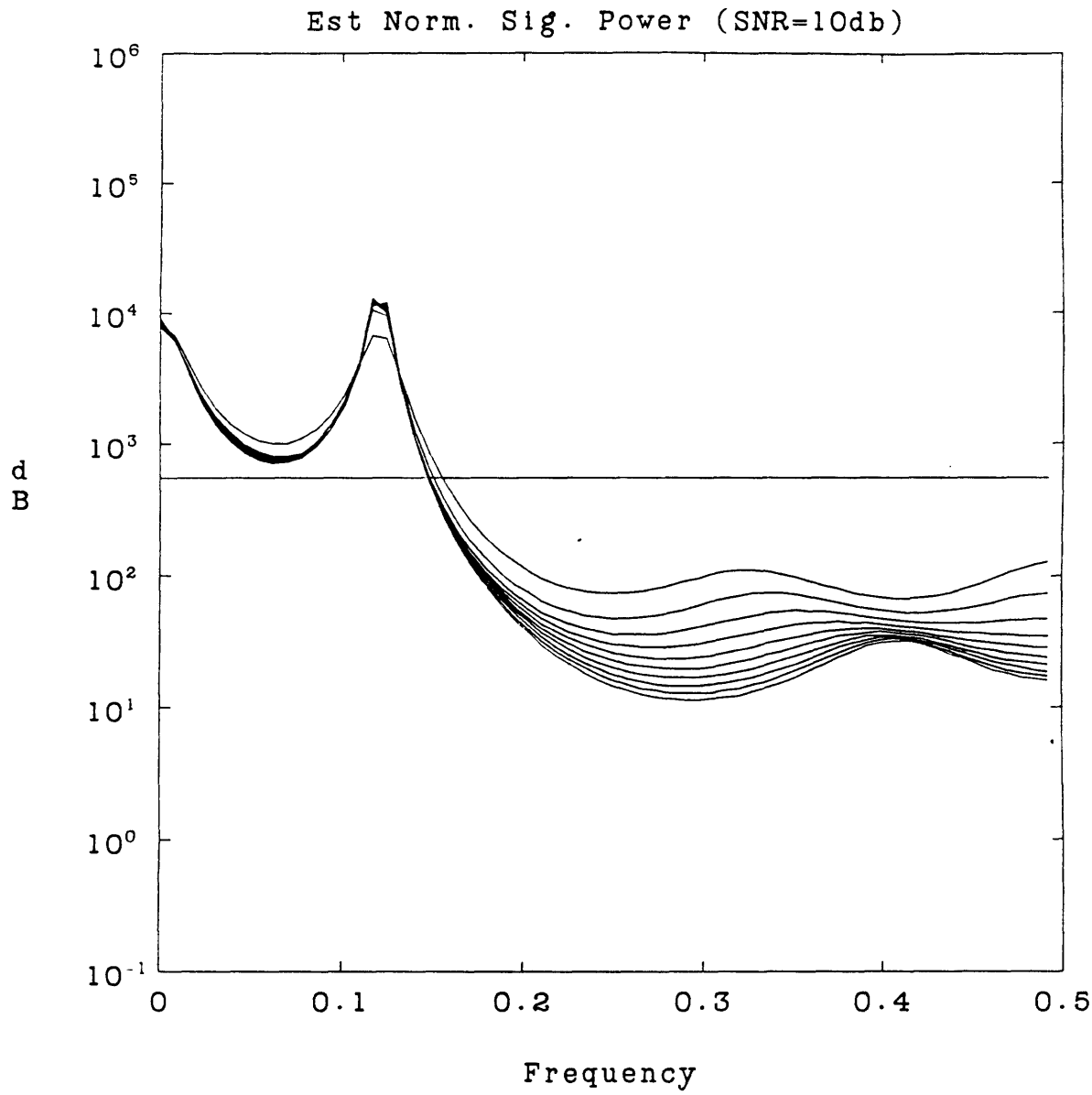


Figure 5g - EM Algorithm, 10 successive estimates of the Normalized Signal Power Spectrum, 1 run, SNR=10dB (dotted is correct value)

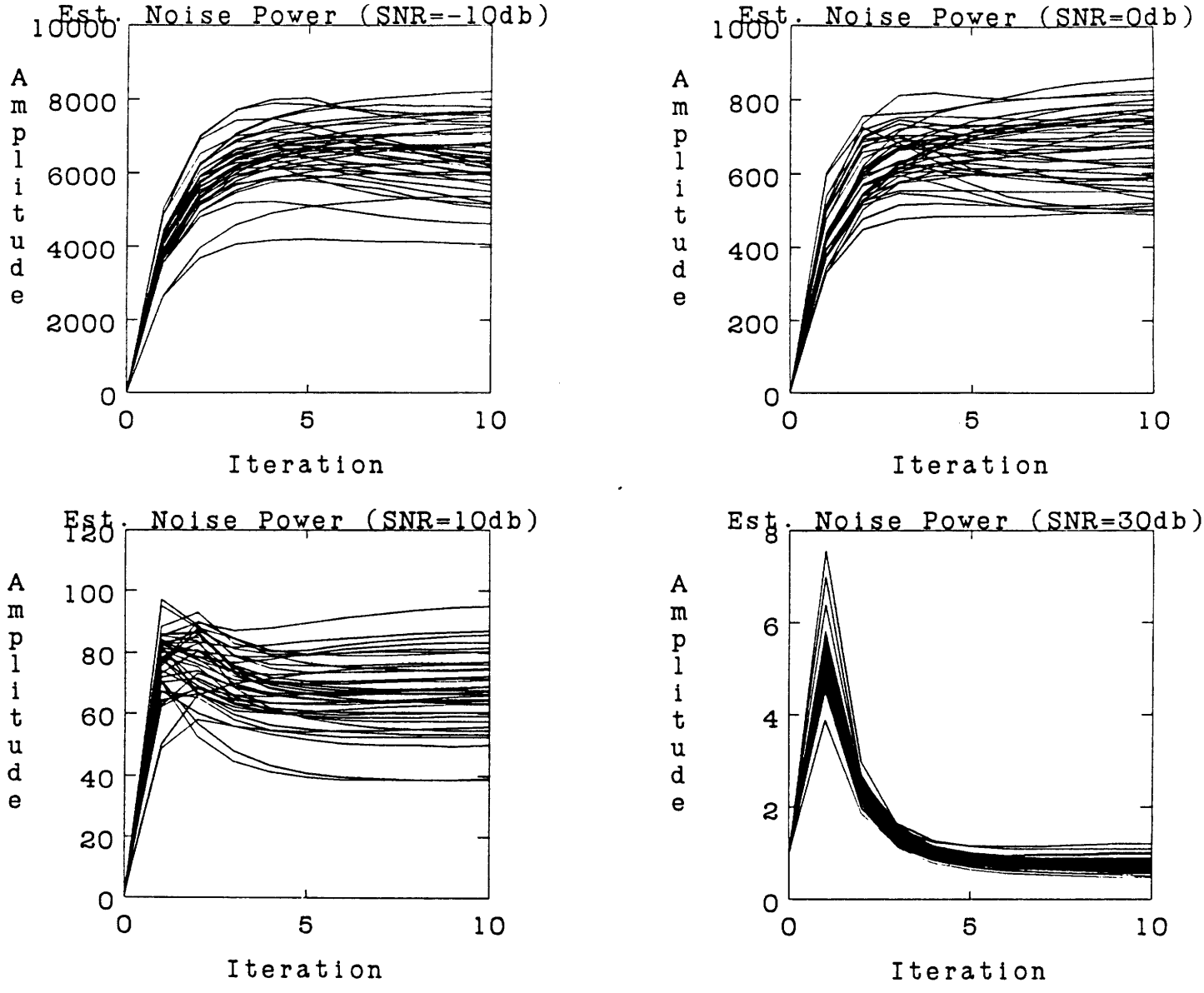


Figure 5h - EM Algorithm, Noise Level Estimates $\hat{\underline{\sigma}}^{(l)}$

solely to capture the relative gain difference between channels. In 5e we show this by plotting the normalized signal gain estimates $\hat{\alpha}_i^{(l)}/\kappa$, where κ is a constant chosen to make the average gain level equal the actual average gain level:

$$\kappa = \left(\frac{\sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}}{\sum_{i=1}^{M} \bar{\alpha}_i^2} \right)^{1/2} \tag{136}$$

The correct values of the normalized signal gains are shown in dotted lines. Note that for SNR above threshold, these normalized signal gains converge rapidly to their correct values.

Figure 5f shows the final 20 power spectra estimates for each of the four SNR levels. Note that while the shape of the spectra are plausible in the low frequency region where the signal energy is greater than the noise energy, the gain of the spectra are too high. This is because the spectral gains compensate for the errors in the signal gains. In figure 5g we plot the estimated total signal energy $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$ compared with the actual total signal energy $P_S(\omega_n; \bar{\underline{\theta}}) \sum_{i=1}^{M} \bar{\alpha}_i^2$. Note that these match quite closely in the low frequency region. In all cases, note that in the high frequency region where noise is stronger than signal, the signal spectral estimates are not correct, and simply drop to about the noise level. Figure 5h shows 10 successive signal power spectrum estimates from 10 iterations with one set of data. Note the rapid convergence at the signal peaks, and the slower convergence in the valleys, exactly as predicted by theory.

Finally, figure 5h shows the convergence of the noise spectral level estimates for both channels. The estimates converge in all cases within about 3 iterations. Note the large initial errors at high SNR caused by poor initial delay estimates.

We also show an example with $M = 3$ channels. The signal and noise spectra are the same as before, and the receiver delays and gains are $\bar{\underline{\tau}} = (-1.58 \;\; 4.29 \;\; 0)^T$ and $\bar{\underline{\alpha}} = (3 \;\; 4.5 \;\; 6)^T$. Initial estimates were generated as in the previous examples. Again we run 20 different sets of data for four different SNR levels, though iterating only 5 times for each run. We estimate all three delays, three signal gains, a 6 pole signal power spectrum with gain g^2, and individual noise spectra levels. Figure 6 plots all these estimates, including the standard deviation of each of the relative delays $\hat{\underline{\tau}}_1^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ and $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_3^{(l)}$. The Cramer-Rao lower bound for the delays is actually a 2×2 matrix.

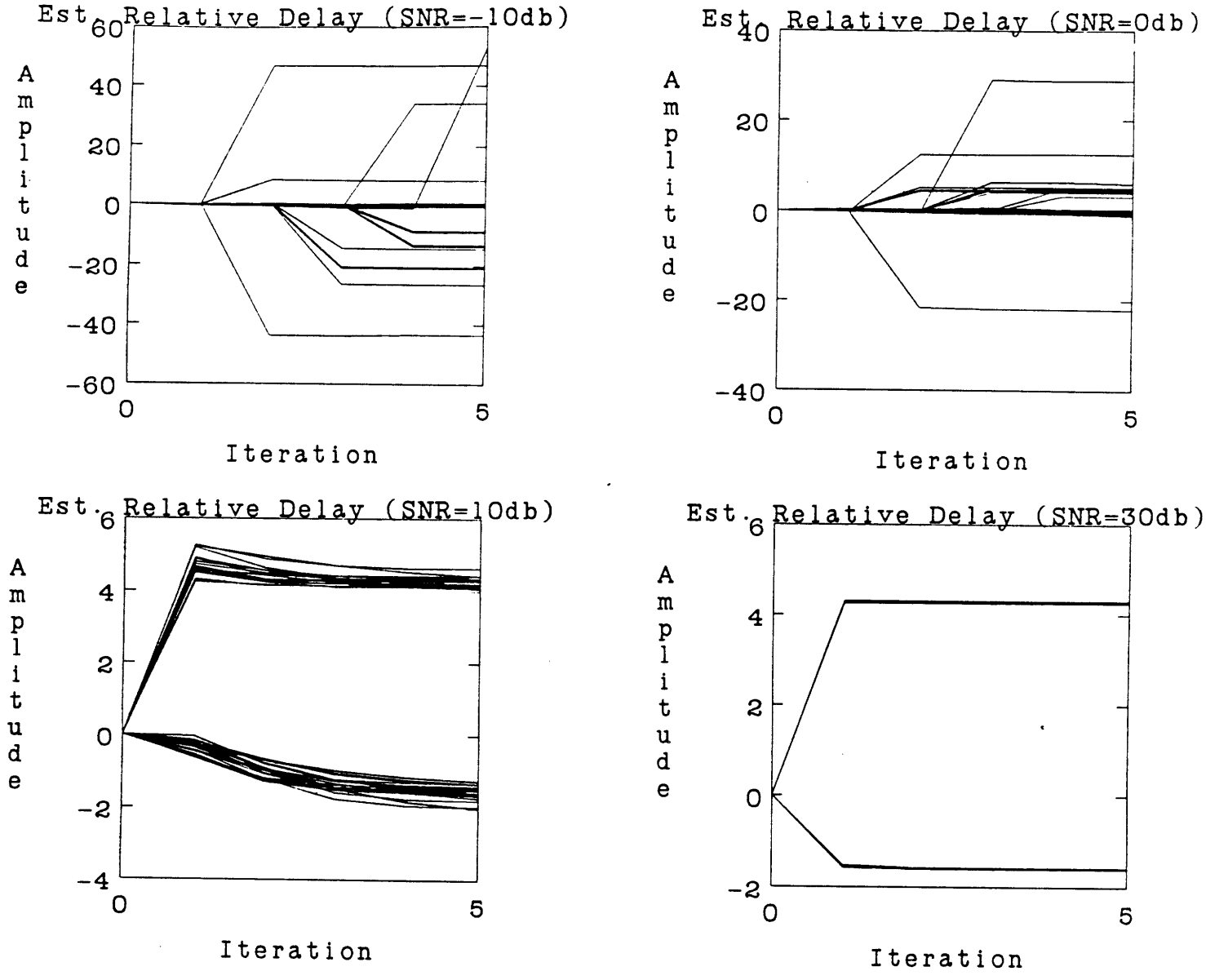


Figure 6a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, 128 point data, $M = 3$ channels, $\underline{\bar{\tau}} = (-1.58 \;\; 4.29 \;\; 0)^T$, $\underline{\bar{\alpha}} = (3 \;\; 4.5 \;\; 6)^T$, 20 runs for each SNR, estimate 3 delays, 3 gains, 6 pole signal power spectrum with gain, 3 noise level parameters.

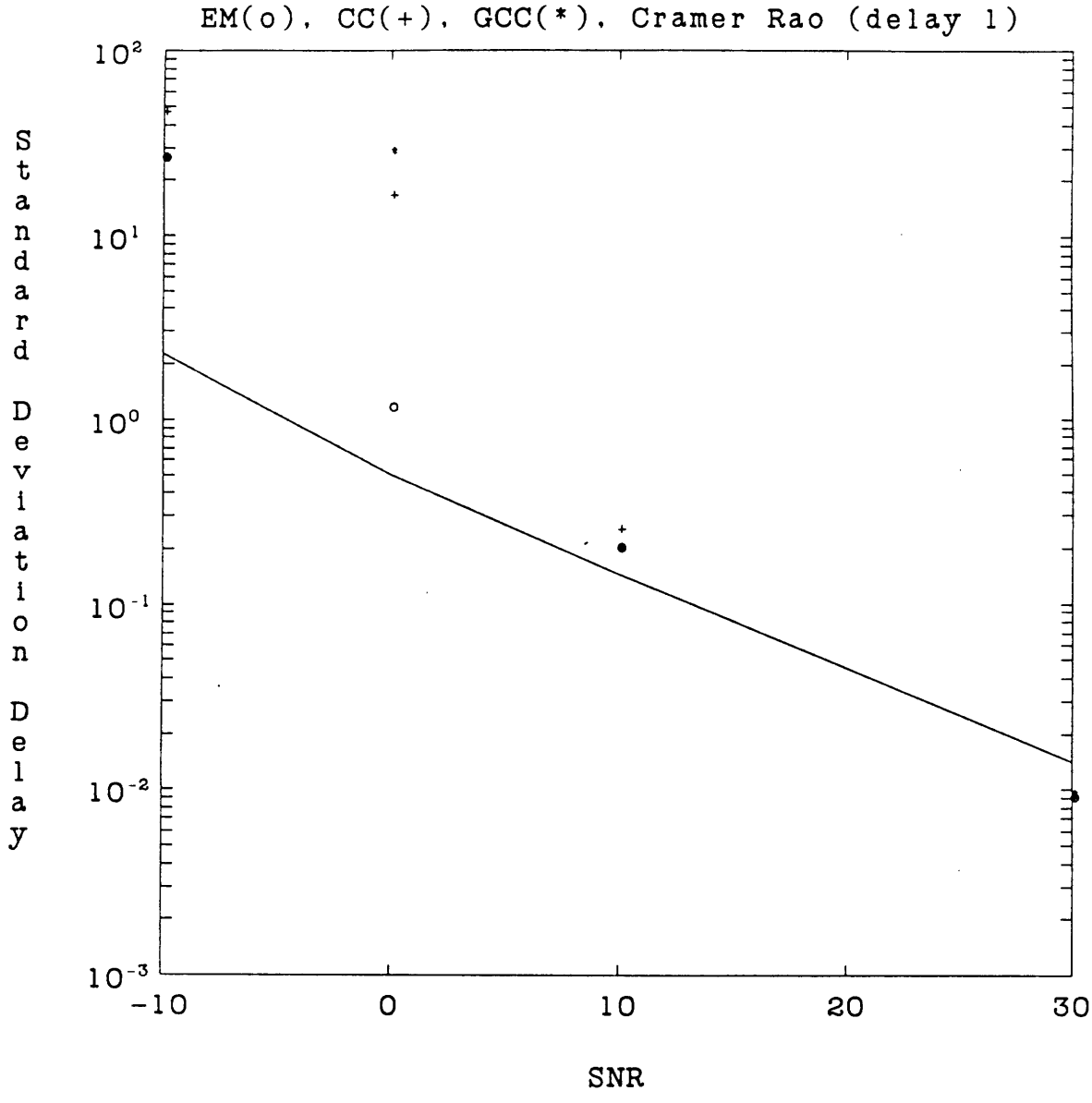


Figure 6b - EM Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_1^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

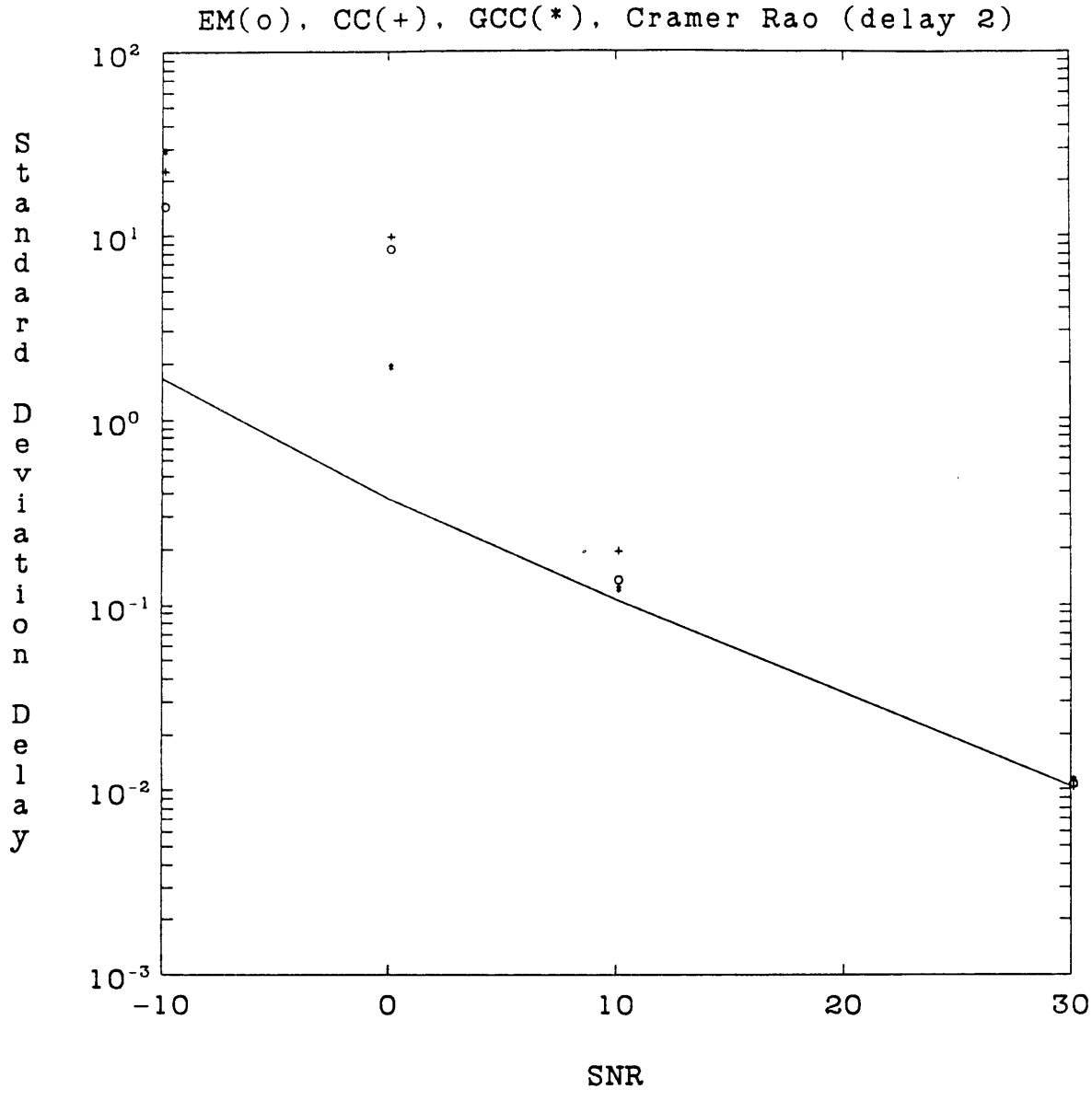


Figure 6c - EM Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

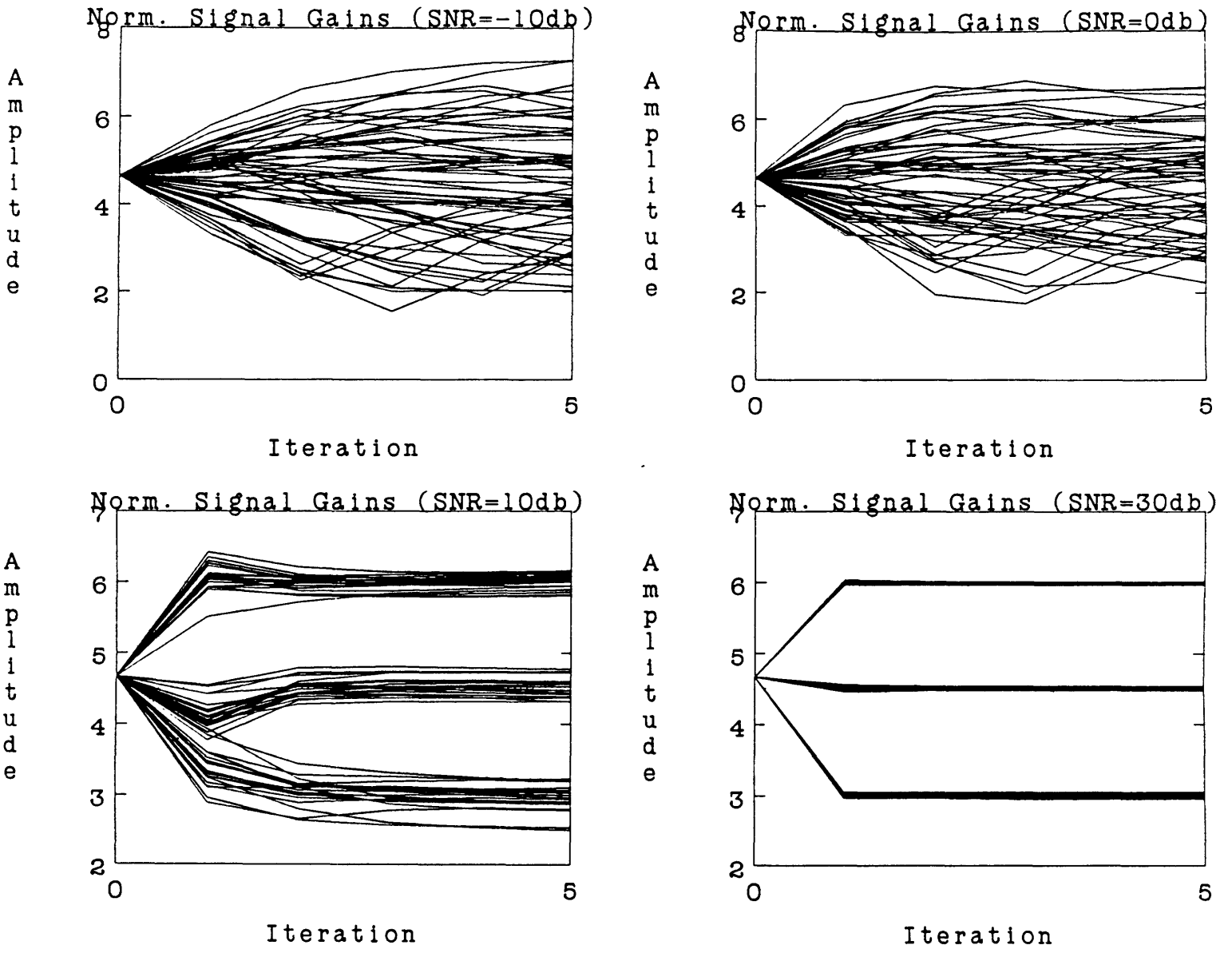


Figure 6d - EM Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

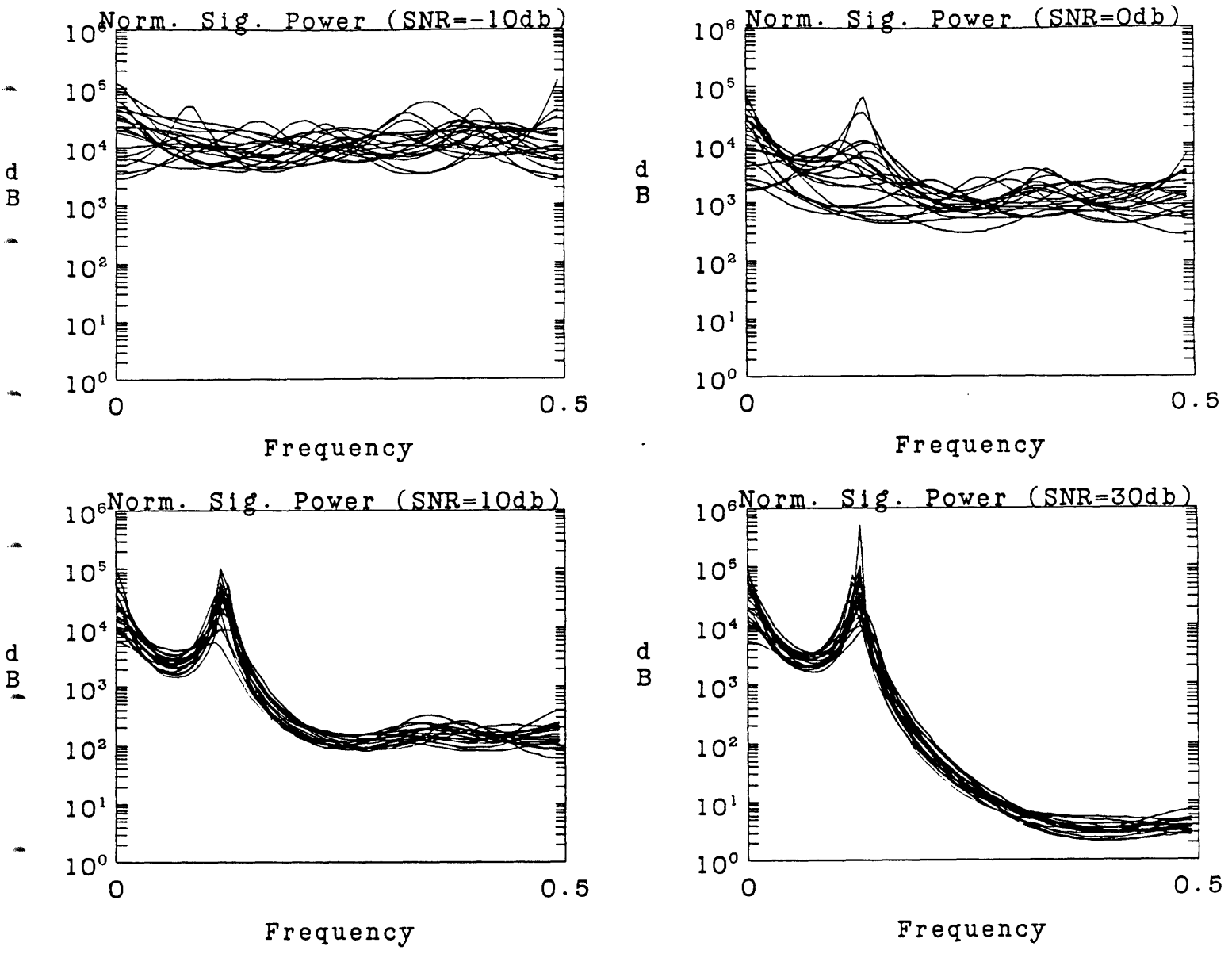


Figure 6e - EM Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

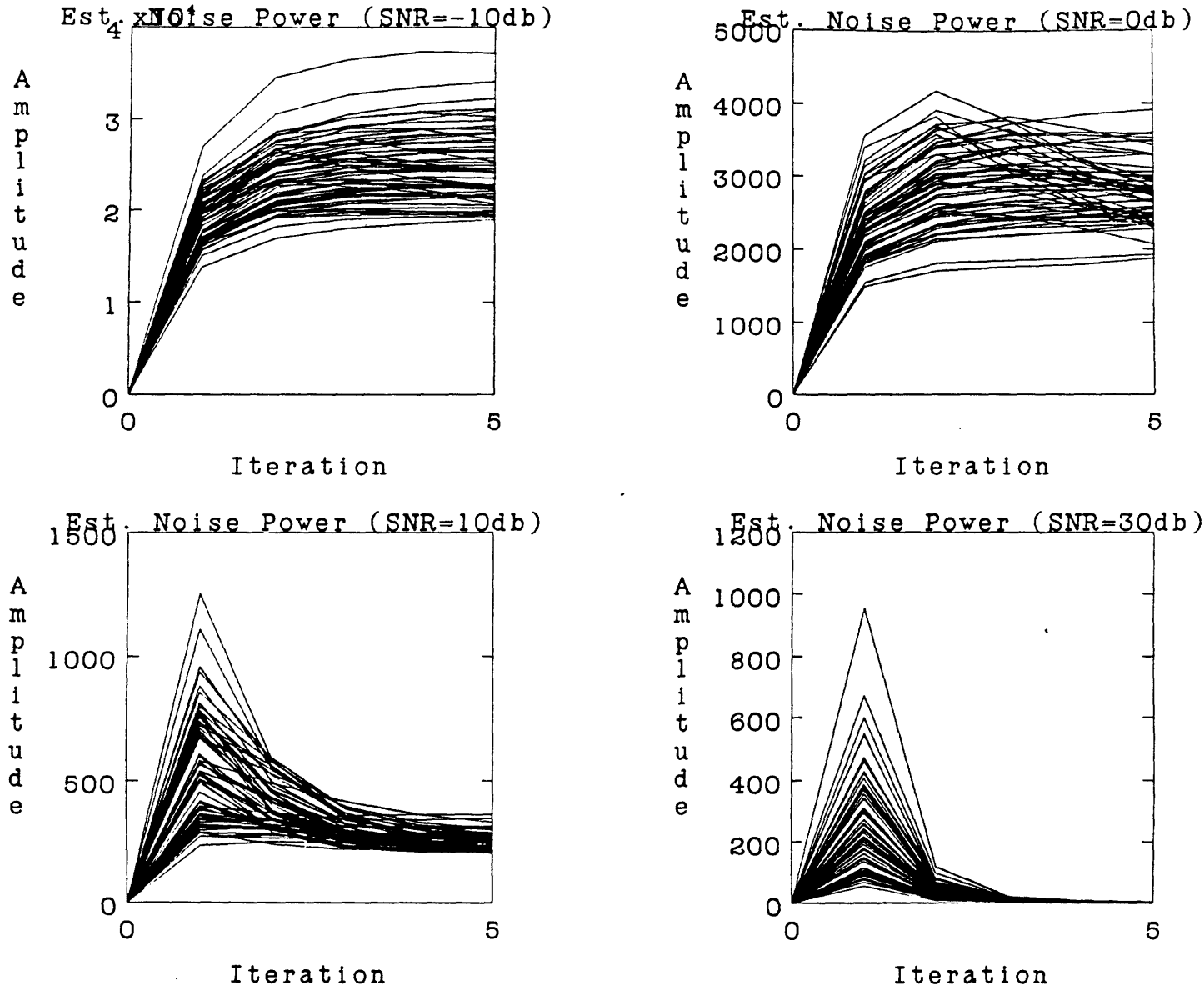


Figure 6f - EM Algorithm, Noise Level Estimates $\hat{\underline{\sigma}}^{(l)}$

The variance of each relative delay, however, can be shown to be bounded below (not tightly) by the corresponding diagonal element of the CR matrix. We therefore plot the standard deviation of each delay as a function of SNR, compared with the square root of the appropriate diagonal element of the CR bound. For completeness, we also plot the standard deviation of delay estimates derived from 3 channel CC and GCC algorithms run on the same data. These latter algorithms use a simple coordinate ascent approach to maximize the two-dimensional unweighted or weighted correlation function.

Note that in the threshold region, SNR=0dB, several large delay estimation errors are made. Note the examples of initial convergence behavior, where estimates suddenly lock onto the right peak, versus asymptotic convergence behavior where estimates converge linearly. Convergence of the delays at high SNR is rapid (1-2 iterations), though as SNR drops, the convergence rate slows. The standard deviation of the final delay estimates is comparable to that of GCC. The signal gains do not converge to the correct average level, due to the gain ambiguity, but the normalized signal gains $\hat{\alpha}_i^{(l)}/\kappa$ do converge within 5 iterations for SNR above threshold. Convergence of the relative gains if fastest at high SNR. The normalized signal power spectra $P_S(\omega_n; \underline{\hat{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$ converge at the peaks to correct values, although at threshold SNR the shapes are often not correct. The noise power estimates converge rapidly, though above threshold SNR the large initial delay estimate errors cause large initial noise level estimate errors.

To further demonstrate the reasonable performance of this algorithm, we present an example with $M = 8$ channels and $N = 256$ points. The signal and noise spectra are the same as before, and the receiver delays and gains are $\underline{\tau} = (\ -4.7\ \ -4.9\ \ -1.5\ \ -3.5\ \ -5.55\ \ -.2\ \ -2.3\ \ 0\ )^T$, and $\underline{\alpha} = (2\ \ 3\ \ 3\ \ 3\ \ 5\ \ 5\ \ 5\ \ 5)^T$. Again we run 20 different sets of data for four different SNR levels: 0dB, 10dB, 20dB, 30dB (n.b. these are higher than in the previous examples). We iterate only 5 times for each run. We estimate all eight delays, eight signal gains, a 10 pole signal power spectrum with gain g^2, and individual noise spectra levels, using the usual initial guesses. Figure 6 plots all these estimates, including the standard deviation of the relative delay $\hat{\tau}_2^{(l)} - \hat{\tau}_8^{(l)}$. The Cramer-Rao lower bound for the delays is actually a 7×7 matrix. The variance

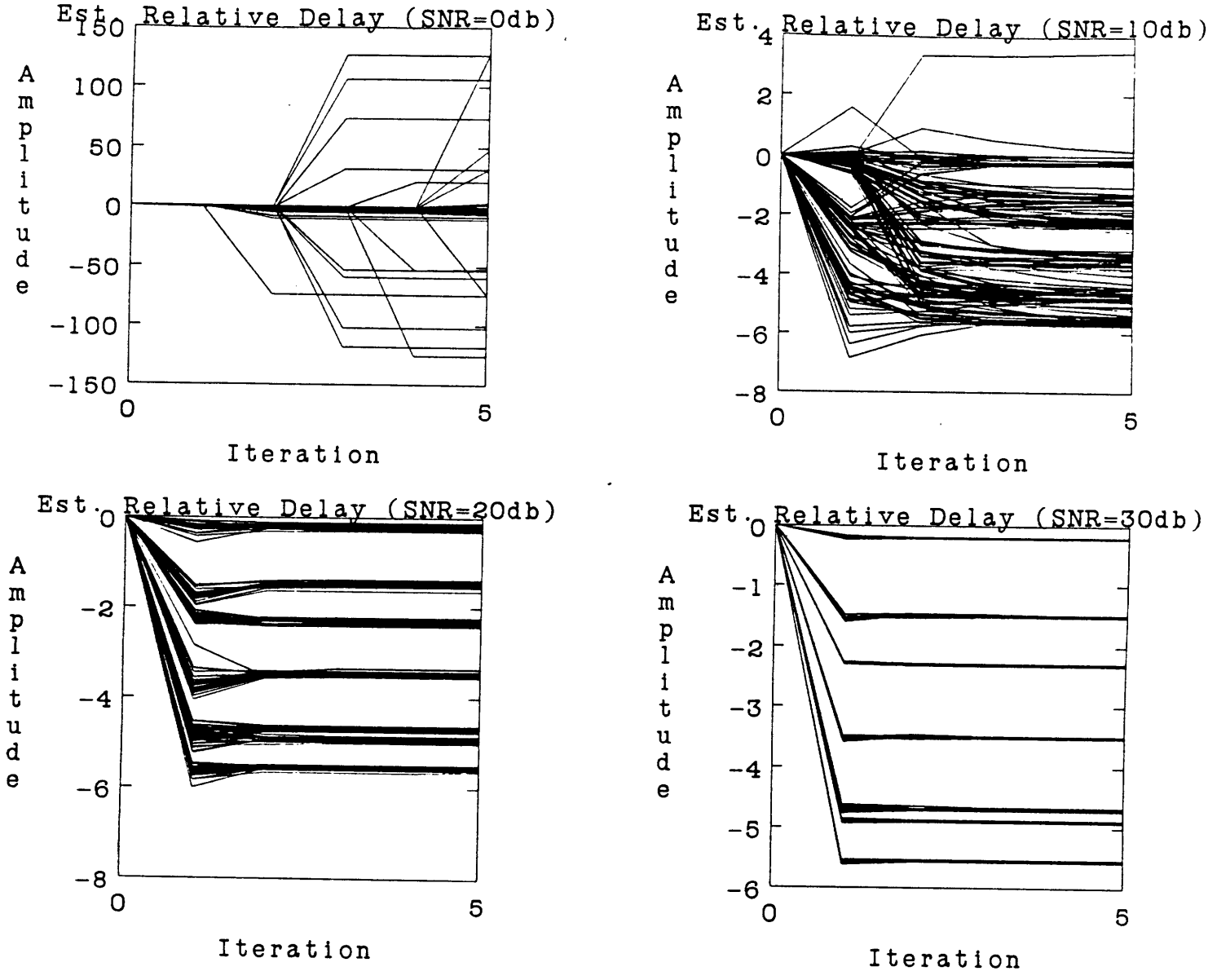


Figure 7a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 256$ point data, $M = 8$ channels, $\underline{\bar{\tau}} = (\ -4.7\ \ -4.9\ \ -1.5\ \ -3.5\ \ -5.55\ \ -.2\ \ -2.3\ \ 0\ )^T$, $\underline{\bar{\alpha}} = (\ 2\ \ 3\ \ 3\ \ 3\ \ 5\ \ 5\ \ 5\ \ 5\ )^T$, 20 runs for each SNR, estimate 8 delays, 8 gains, 10 pole signal power spectrum with gain, 8 noise level parameters.

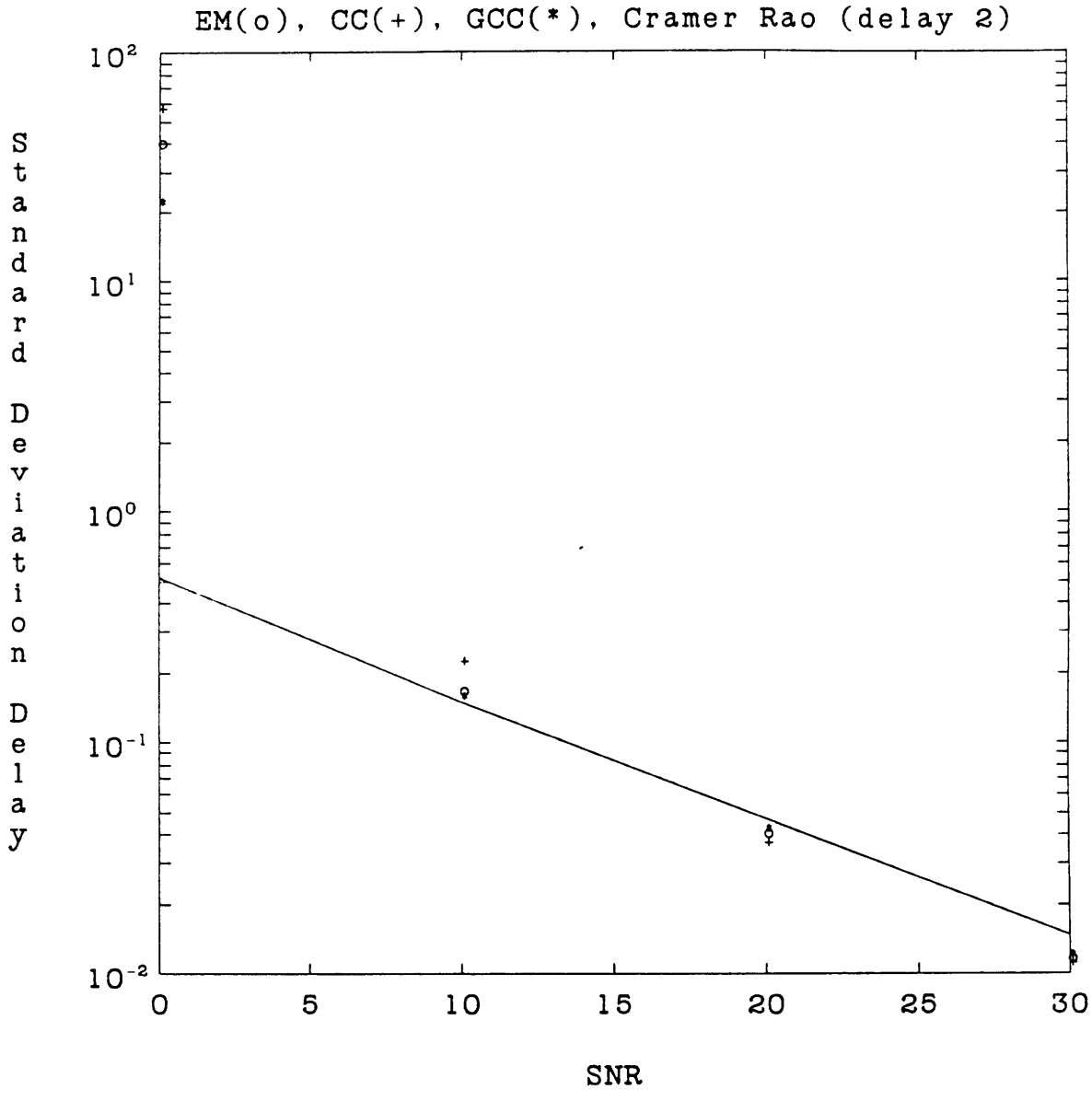


Figure 7b - EM Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_8^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

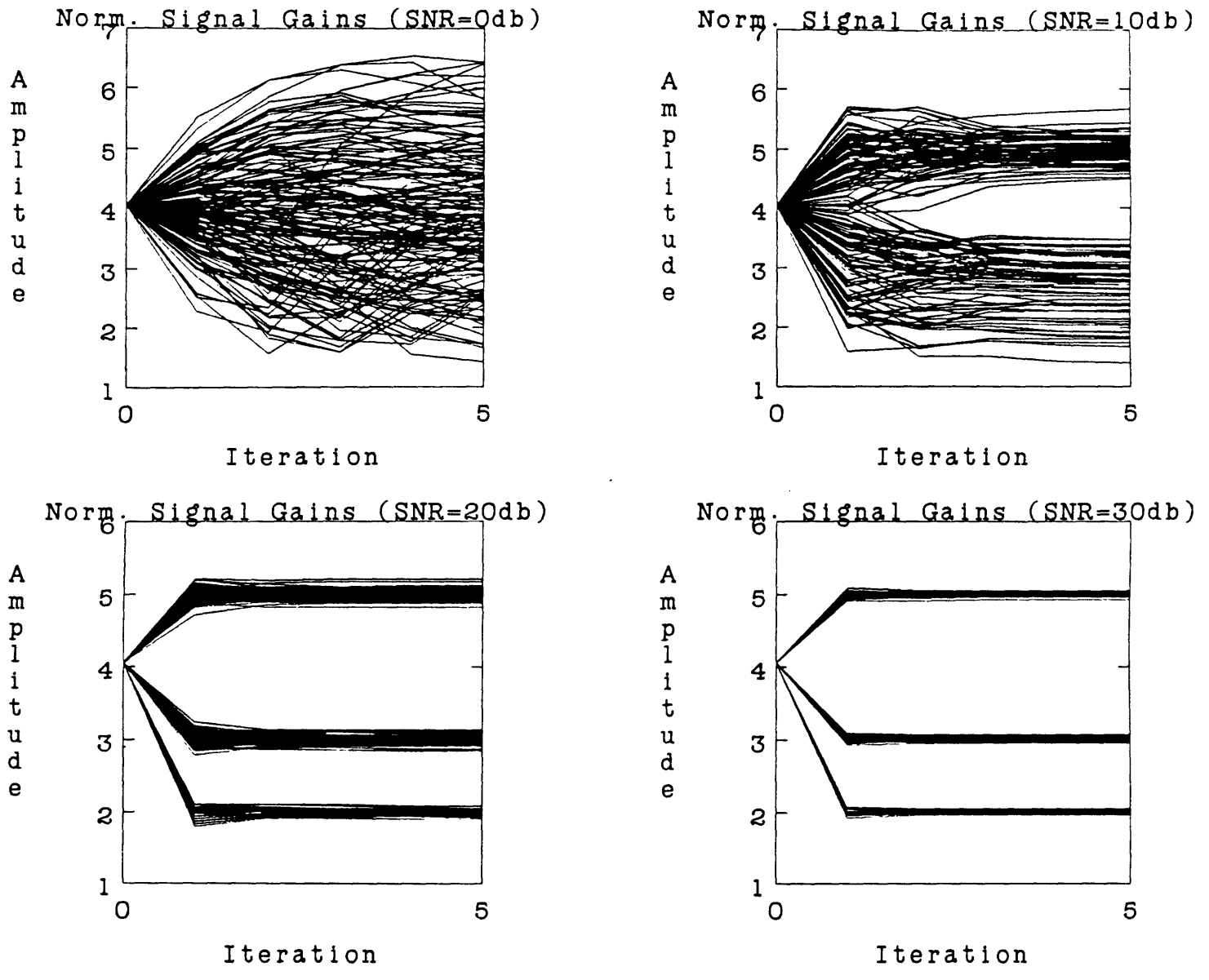


Figure 7c - EM Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

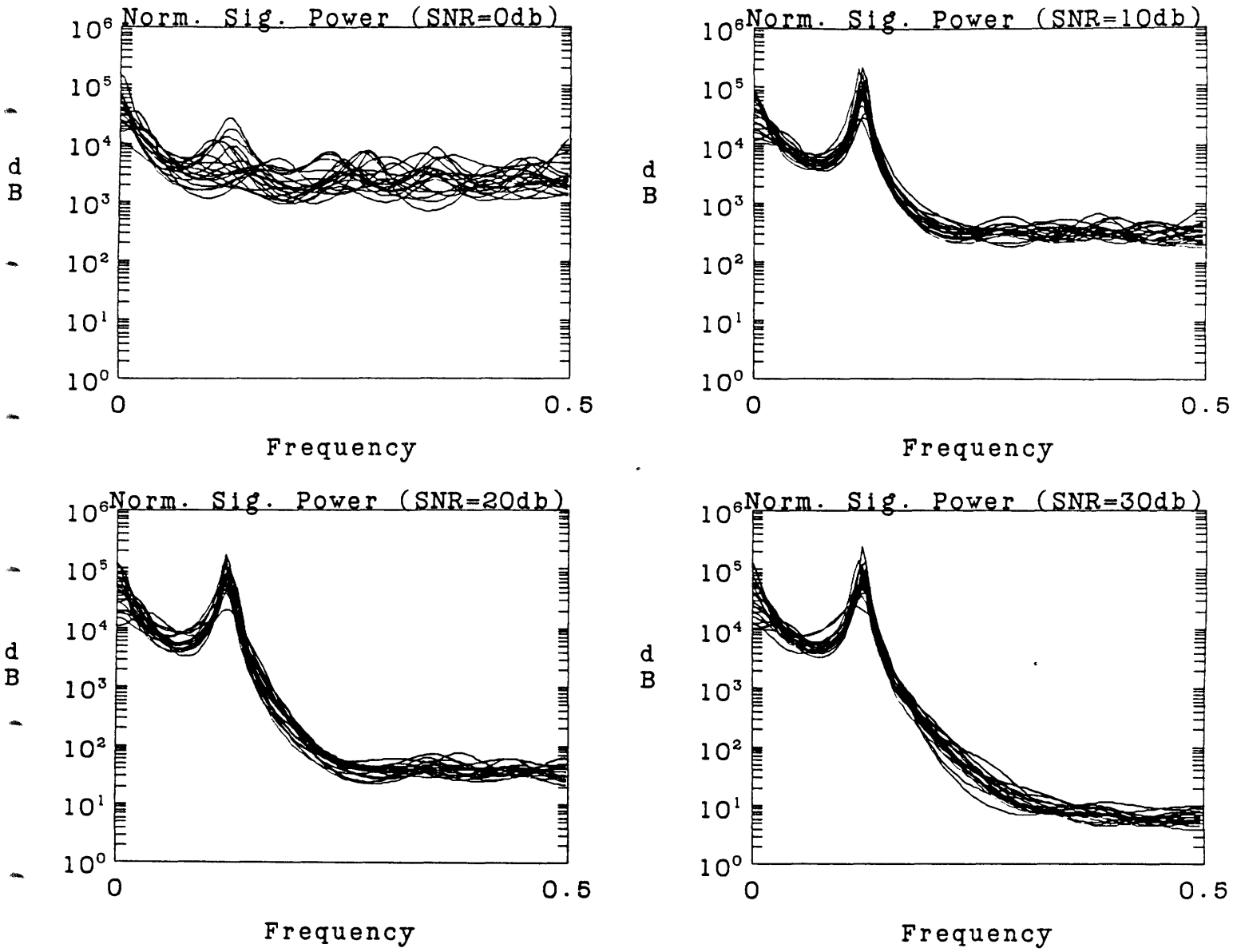


Figure 7d - EM Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

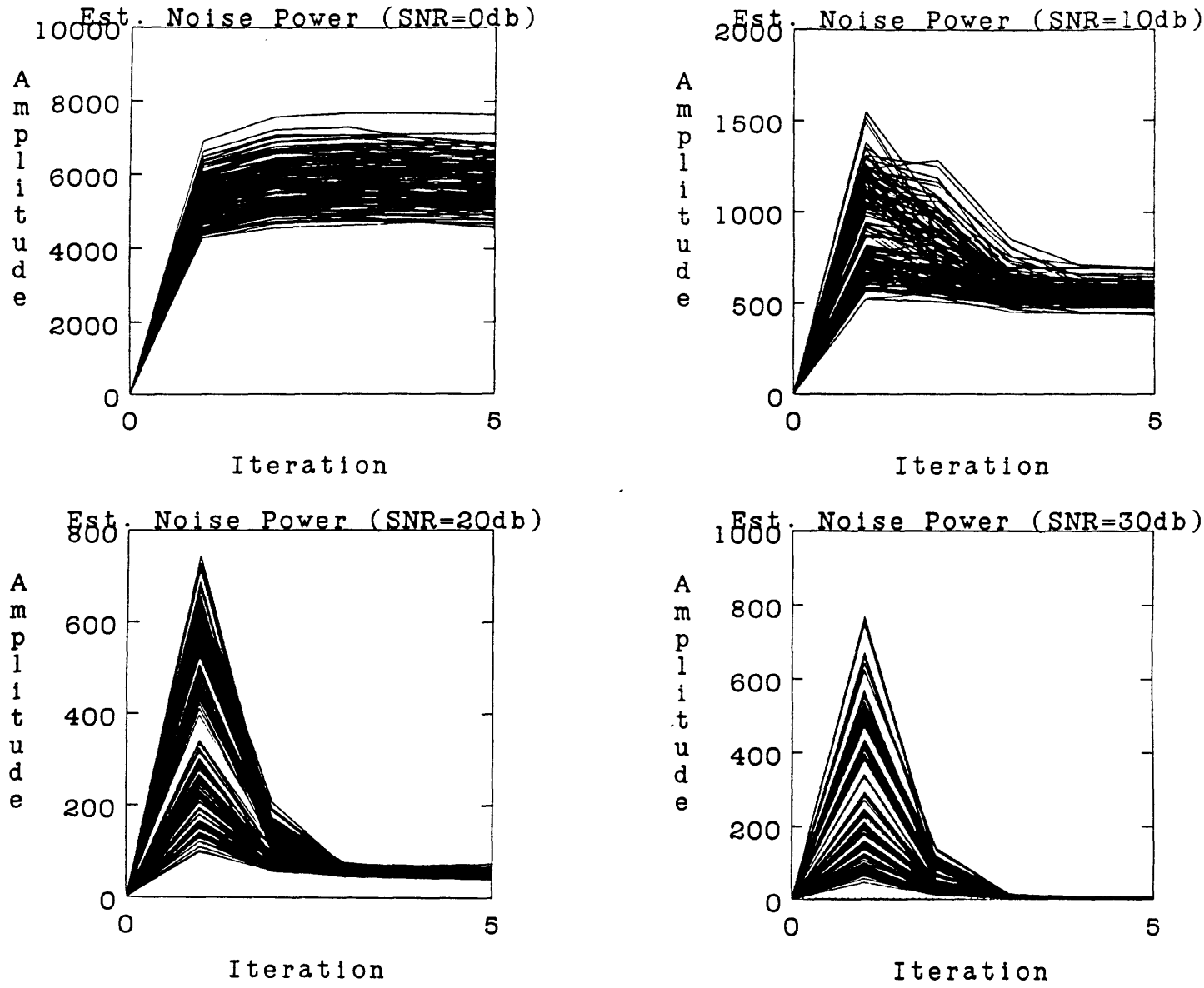
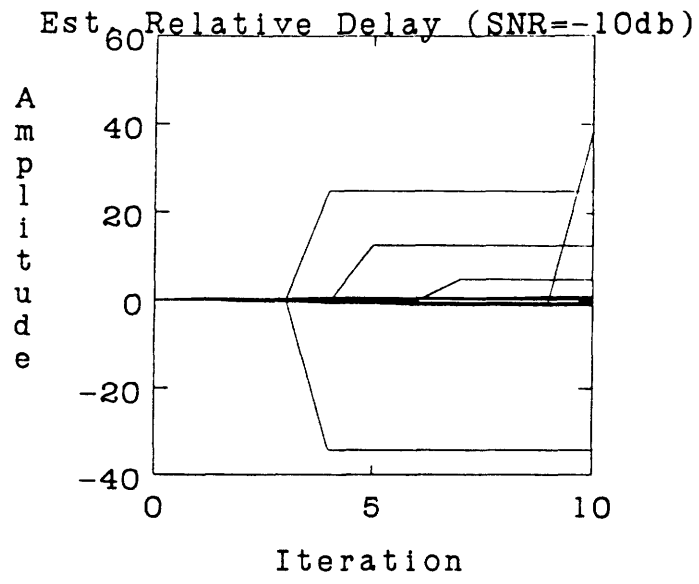


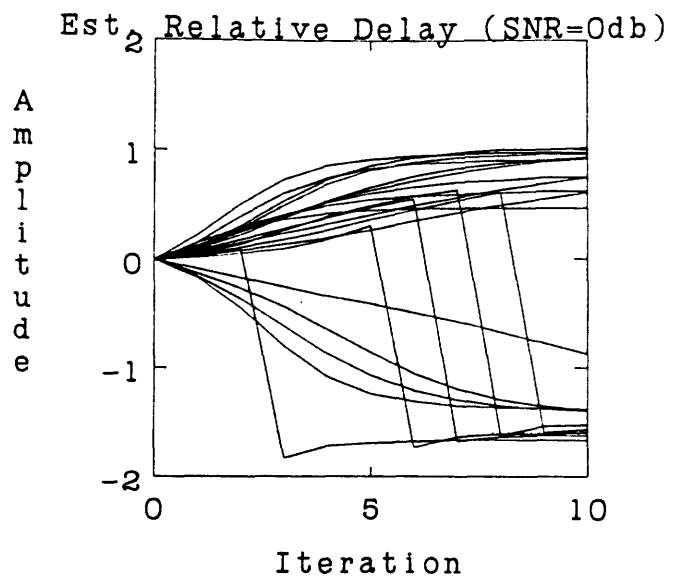
Figure 7e - EM Algorithm, Noise Level Estimates $\hat{\underline{\sigma}}^{(l)}$

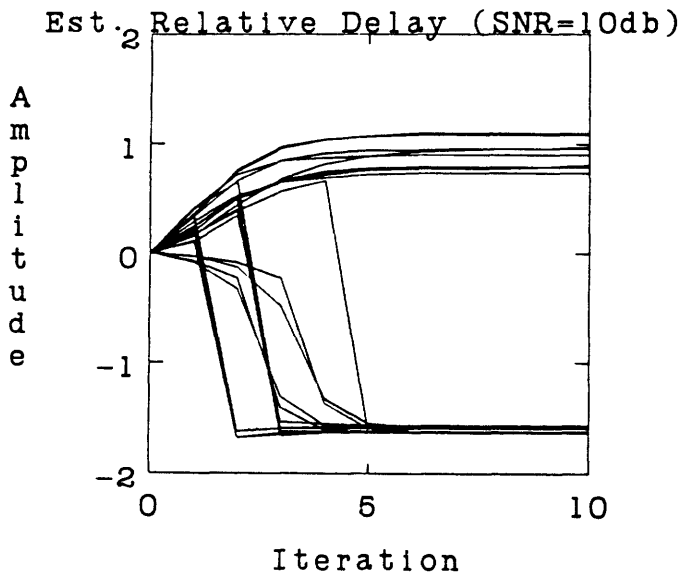
of each relative delay, however, can be shown to be bounded below (not tightly) by the corresponding diagonal element of the CR matrix. We therefore plot the standard deviation of the delay as a function of SNR, compared with the square root of the appropriate diagonal element of the CR bound. For completeness, we also plot the standard deviation of delay estimates derived from 8 channel CC and GCC algorithms, on the same data. These latter algorithms use a simple coordinate ascent approach to maximize the seven-dimensional unweighted or weighted correlation function.

Note that in the threshold region, SNR=0dB, several large delay estimation errors are made. Convergence of the delays above threshold SNR is rather rapid (about 2 iterations). Despite the lack of knowledge of the correct model, the standard deviation of the final delay estimates is comparable to that of GCC. The signal gains do not converge to the correct average level, due to the gain ambiguity, but the normalized signal gains $\hat{\alpha}_i^{(l)}/\kappa$ converge rapidly (1-3 iterations) for SNR=10dB and above. The normalized signal power spectra $P_S(\omega_n;\underline{\hat{\theta}}^{(l)})\sum_{i=1}^{M}\hat{\alpha}_i^{(l)^2}$ converge at the peaks to correct values, although at threshold SNR the shapes are often not correct. The noise power estimates converge rapidly, though above threshold SNR the large initial delay estimate errors cause large noise level estimate errors after the first iteration.

We conclude this section with an example in which the EM algorithm fails. Consider a 2 channel system with the delays (-1.58, 0) and gains (3,3), but with a white signal and white noise. We estimate just the two delays and the two gains, and use the correct signal and noise power spectra. Figure 8 plots the delay estimates for 20 runs at four SNR levels. Note that substantial numbers of incorrect delay estimates are found, even at SNR=30dB! The problem is that both the initial gain estimates and the true gain values are equal to each other. As pointed out by our theoretical analysis, each function that EM maximizes for each delay has two peaks, one at the correct delay value and one at the previous estimate. With equal signal energy in both channels, the peaks at the old estimates are larger than the peaks at the correct value. Given some noise, the EM method tends to get confused as to which peak to pick, and can lock onto the wrong one. As we will see, this problem is eliminated by the EM-ML delay estimation algorithm.







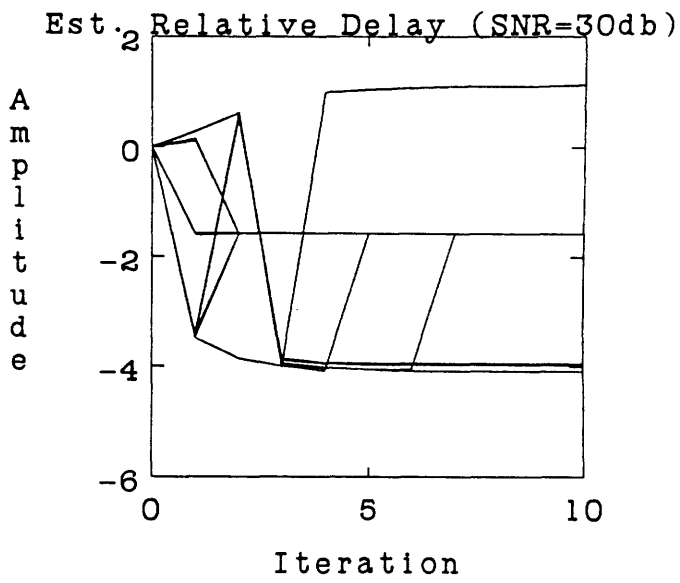


Figure 8a - EM Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known flat (white) signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains.

7.3 EM-ML Individual Delay Estimation

We ran similar simulations to test the behavior of the EM-ML individual delay estimation algorithm. In this approach, the delays are estimated one at a time by maximizing the likelihood function directly over each delay in turn. EM steps are then used to estimate the remaining parameters. We used exactly the same signal and noise models as in the previous section, using the same data sequences.

Figure 9 shows the behavior in the $M = 2$ example when we estimate the two time delays together with both signal gains. We assume that the signal and noise spectra are known. We use the same initial guesses as before. Compare these figures with the EM algorithm in 3. Note that the EM-ML delay estimates converge quickly to their final values. At SNR=0dB, EM-ML makes large initial delay errors twice, and cannot recover in one of the cases. This is typical behavior near the threshold. (The flawless performance of EM at SNR=0dB was a statistical fluke). Delay estimate standard deviation is similar to the GCC method. The signal gain estimates, as usual, converge very slowly.

Figure 10 shows the behavior when we estimate the all the parameters: two time delays, plus both signal gains, a 6 pole signal spectrum with gain g^2, and individual noise levels on the two receivers. Note that the delay estimates converge somewhat more slowly to their final values (about 2 iterations), due to the initial use of poor spectral estimates. In the threshold case, SNR=0dB, more initial large delay estimate errors are made than when the spectra are known. Compared with figure 5, convergence is significantly faster than EM, except at 30dB where both methods essentially converge in 1-2 iterations. The standard deviation of the final delay estimates is still similar to that of GCC. The gains do not converge to their correct values, due to the gain ambiguity discussed earlier. However, the normalized gain estimates do converge within a couple iterations, with fastest convergence at high SNR. The estimated signal power spectra converge quickly near the peaks, but are poor in the valleys where the signal level dips below the noise floor. Finally, the noise power level estimates are comparable to those of EM, though they converge faster due to the faster delay estimate convergence.

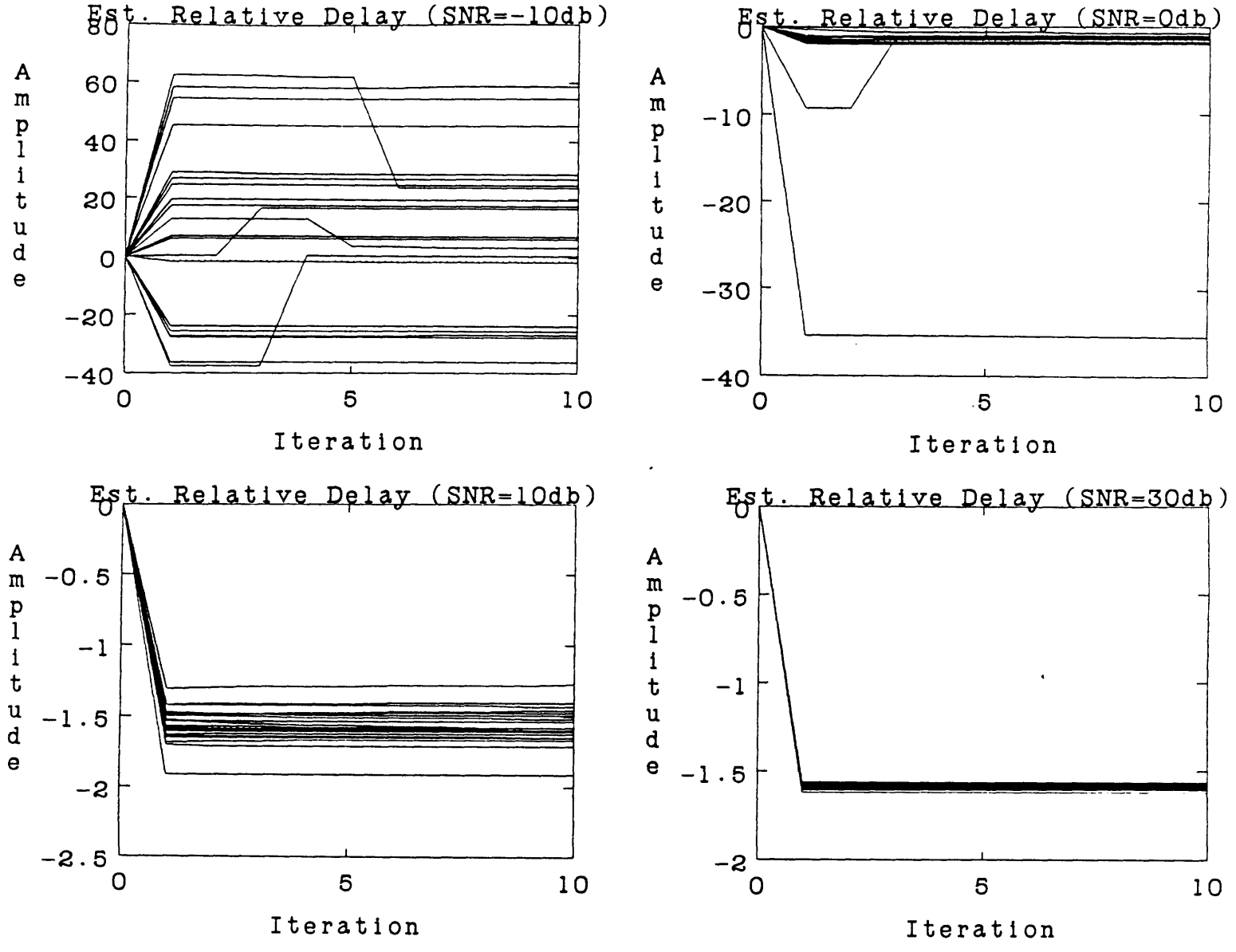


Figure 9a - EM-ML Delay Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known lowpass signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains.

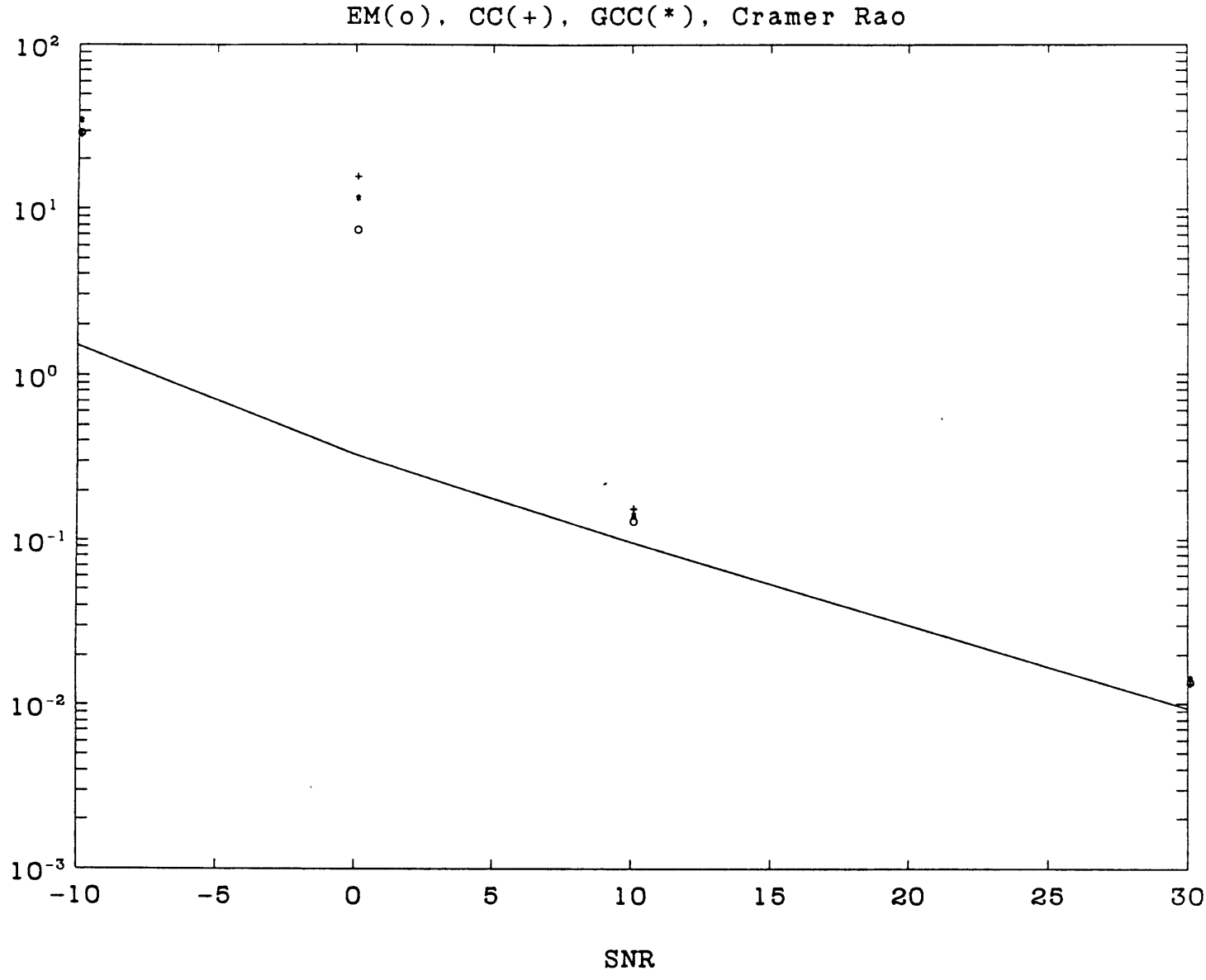


Figure 9b - EM-ML Delay Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

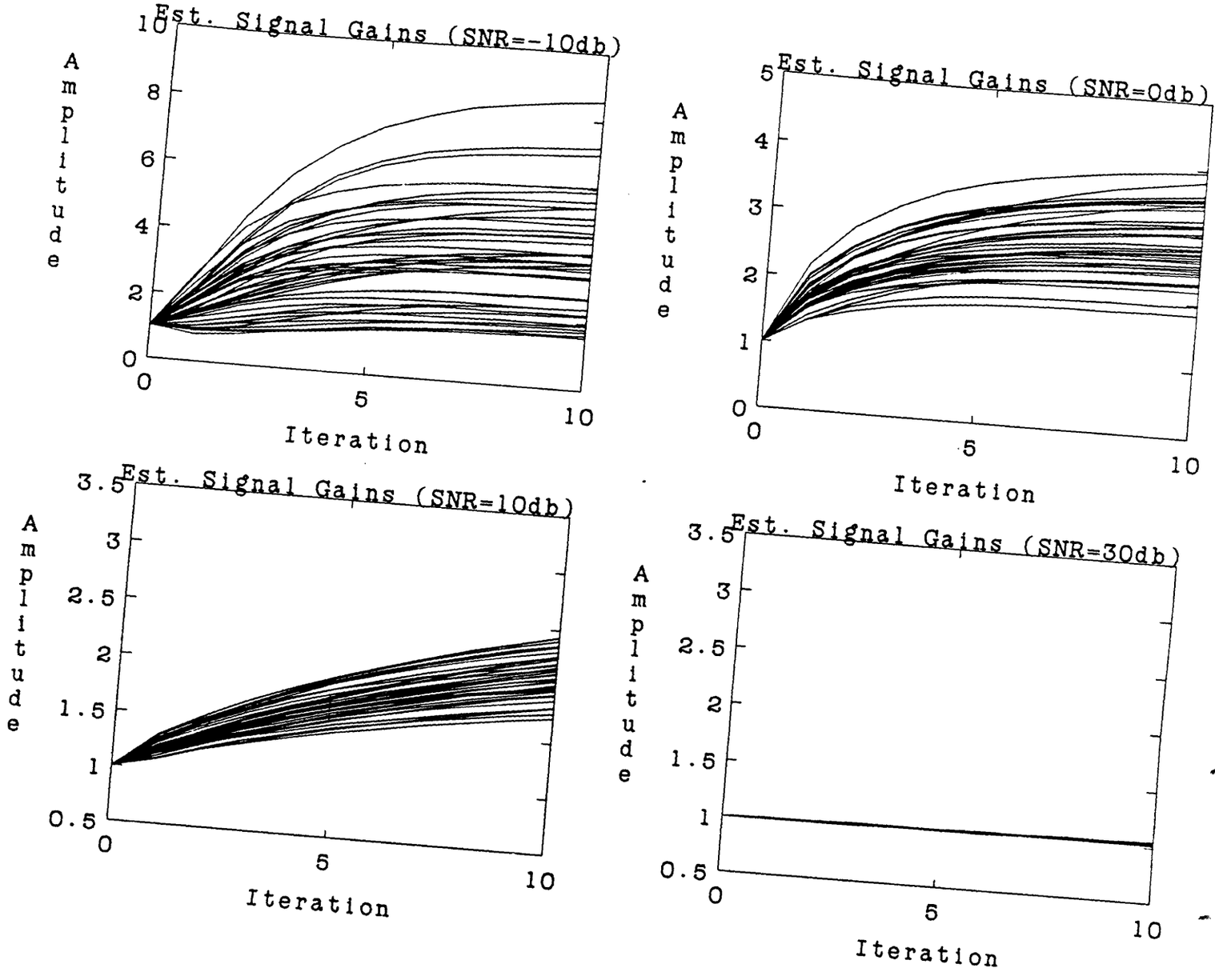
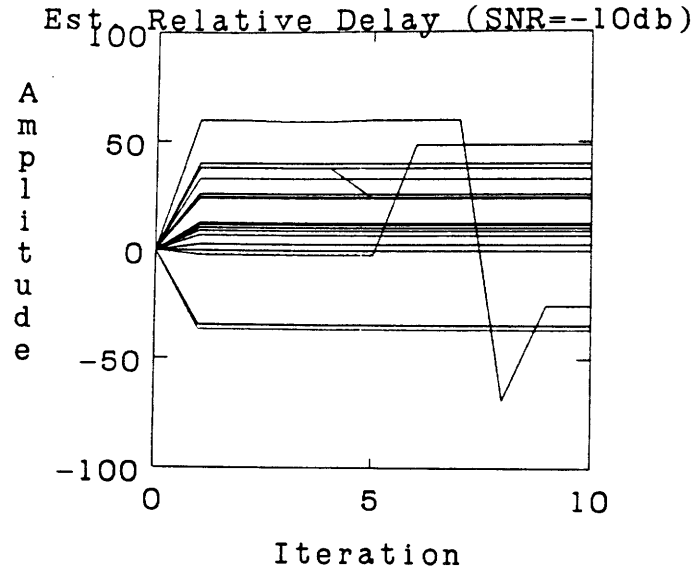
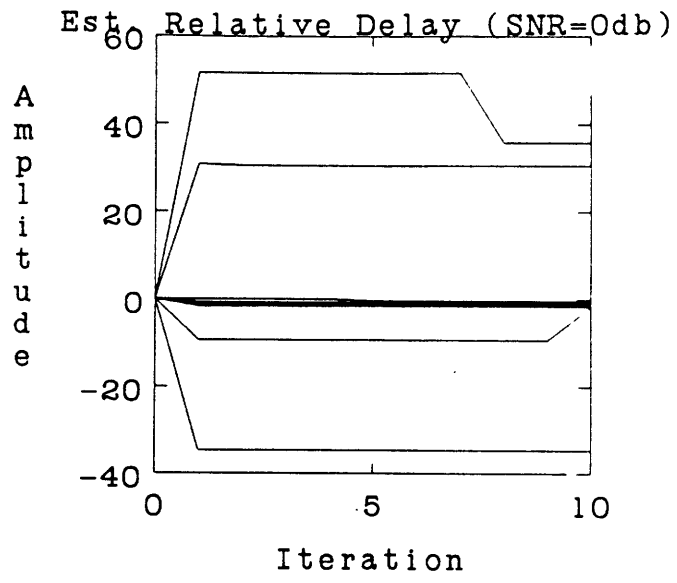
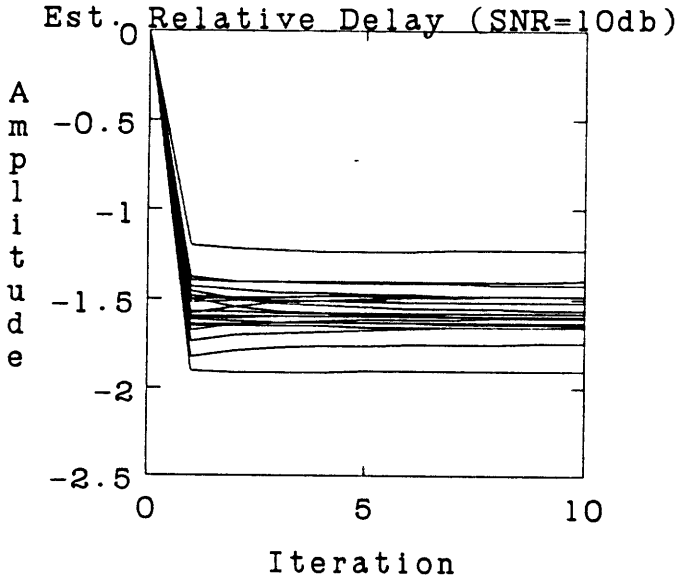


Figure 9c - EM-ML Delay Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$







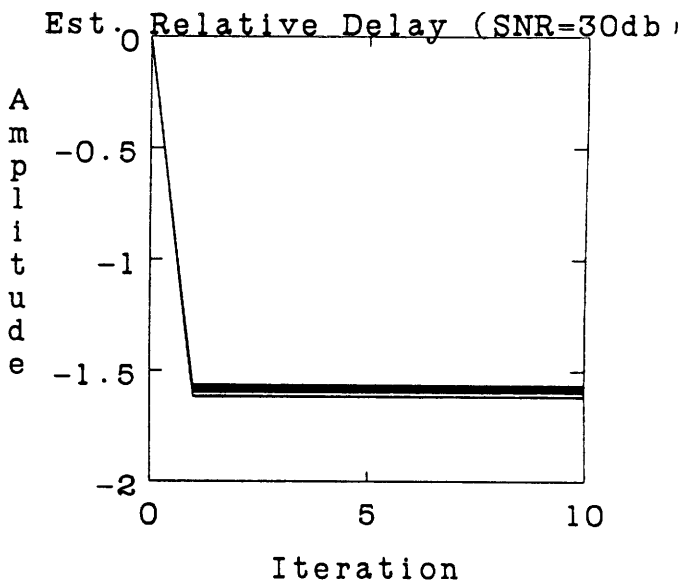


Figure 10a - EM-ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3\ \ 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains, 6 pole signal power spectrum with gain, 2 noise level parameters.

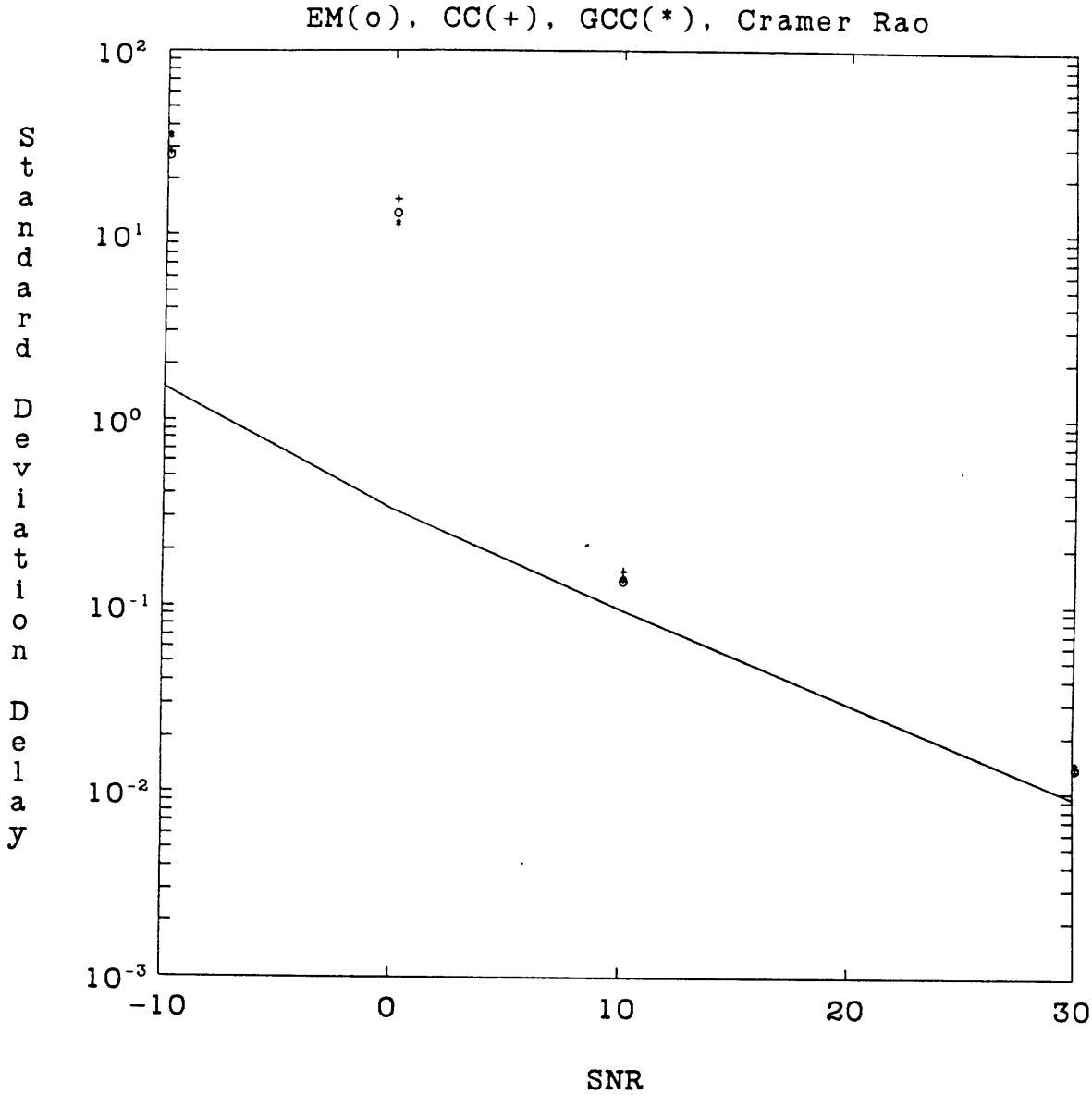


Figure 10b - EM-ML Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

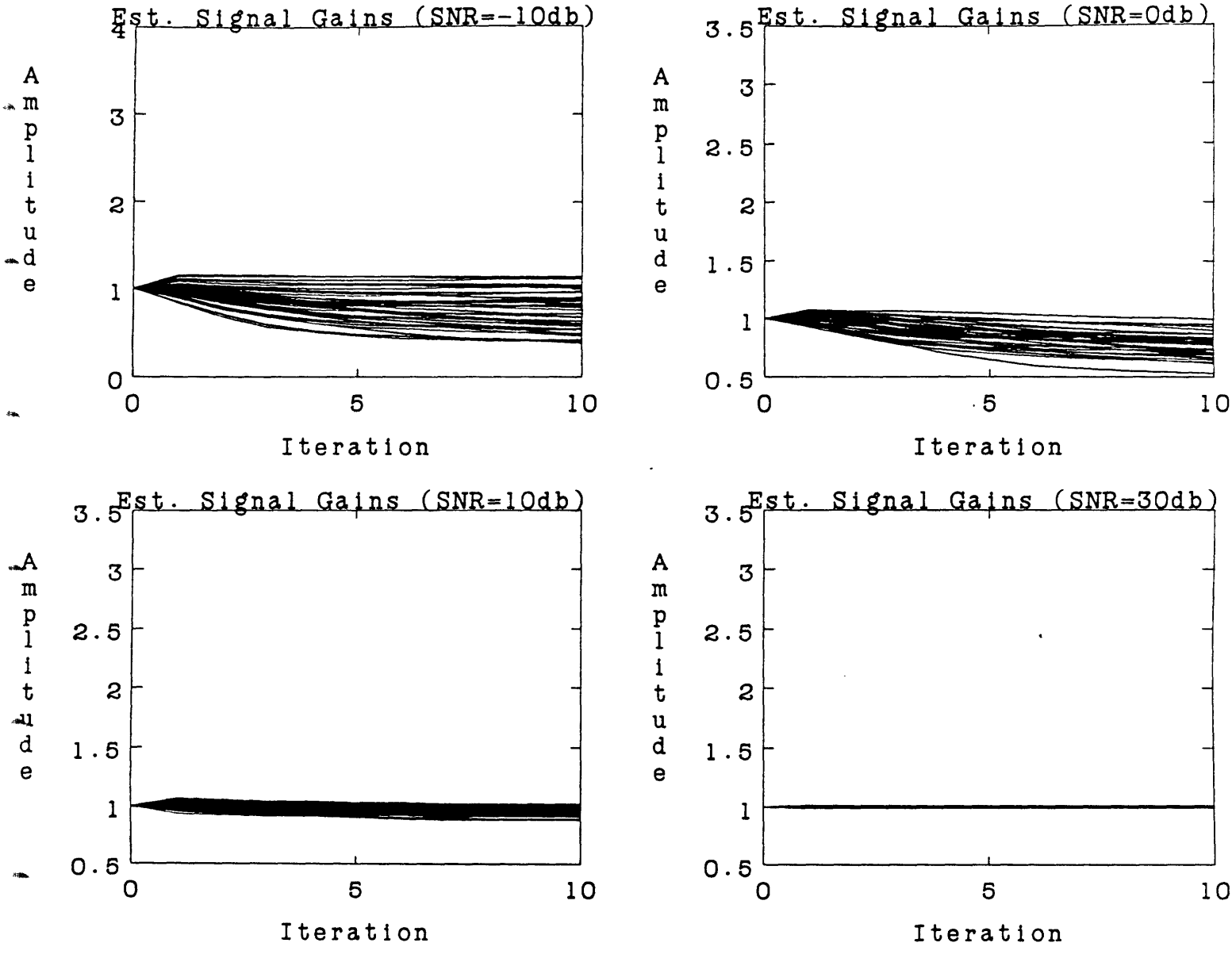


Figure 10c - EM-ML Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$.

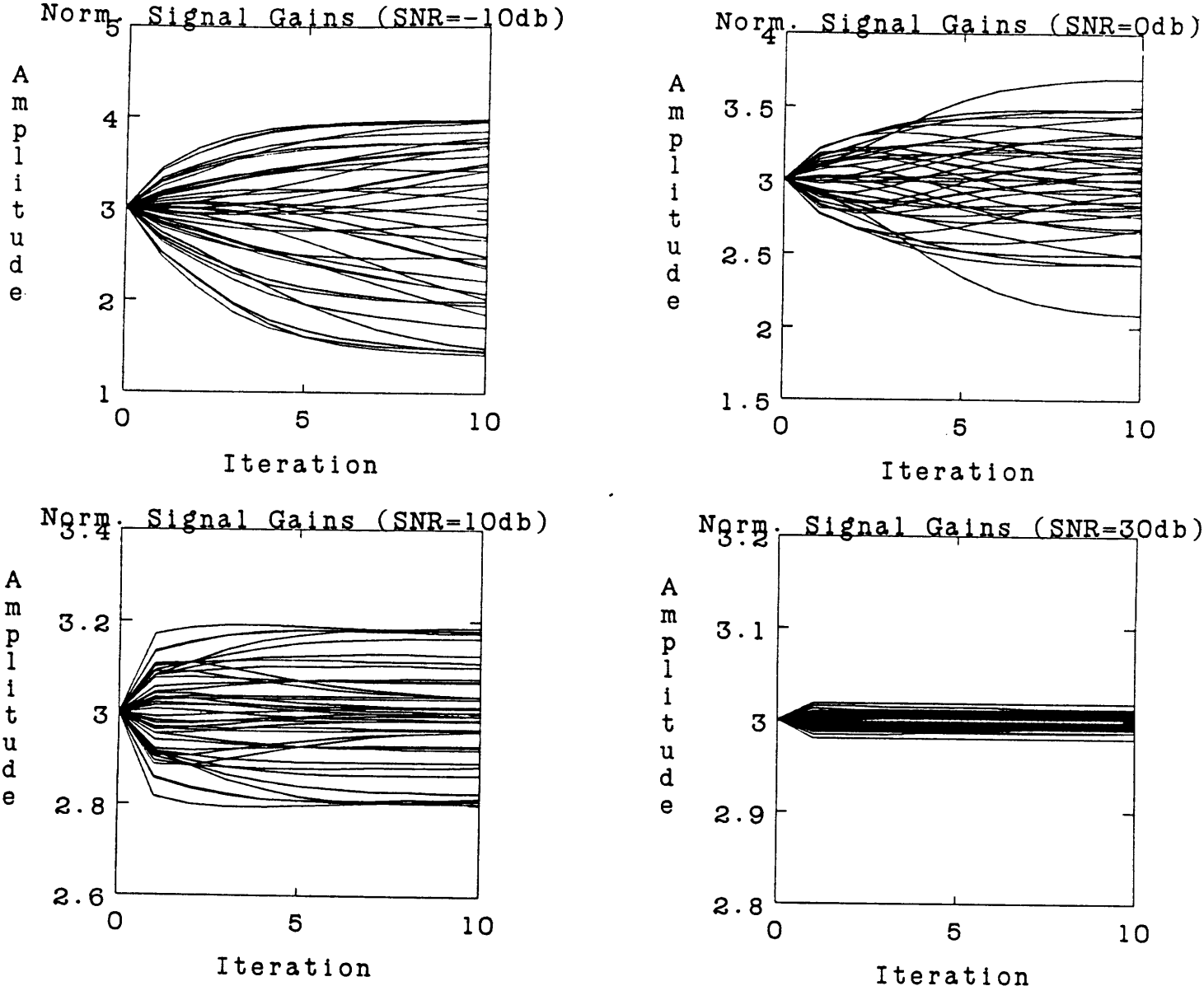


Figure 10d - EM-ML Algorithm, Normalized Gain Estimates $\underline{\hat{\alpha}}^{(l)}/\kappa$.

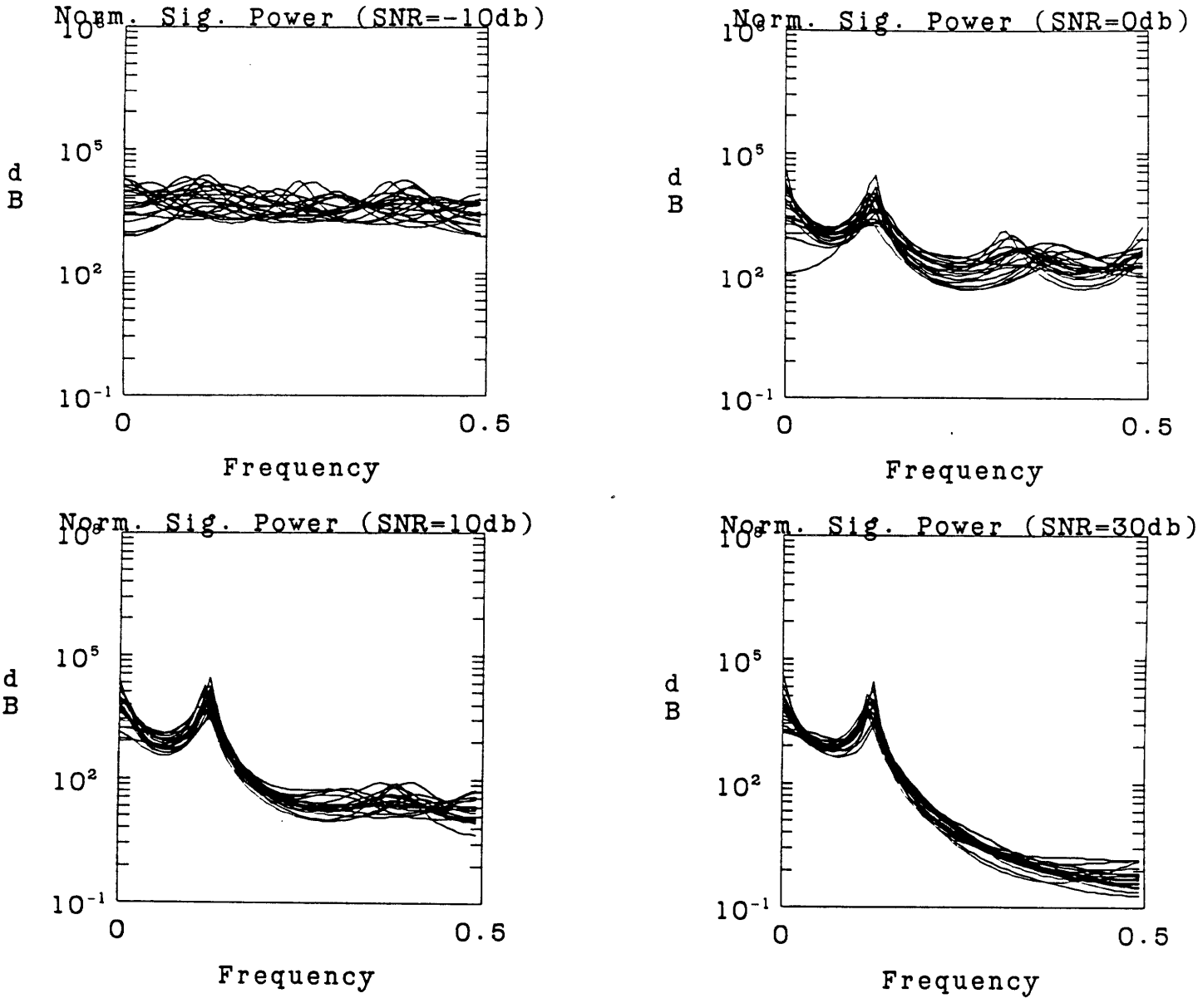


Figure 10e - EM-ML Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

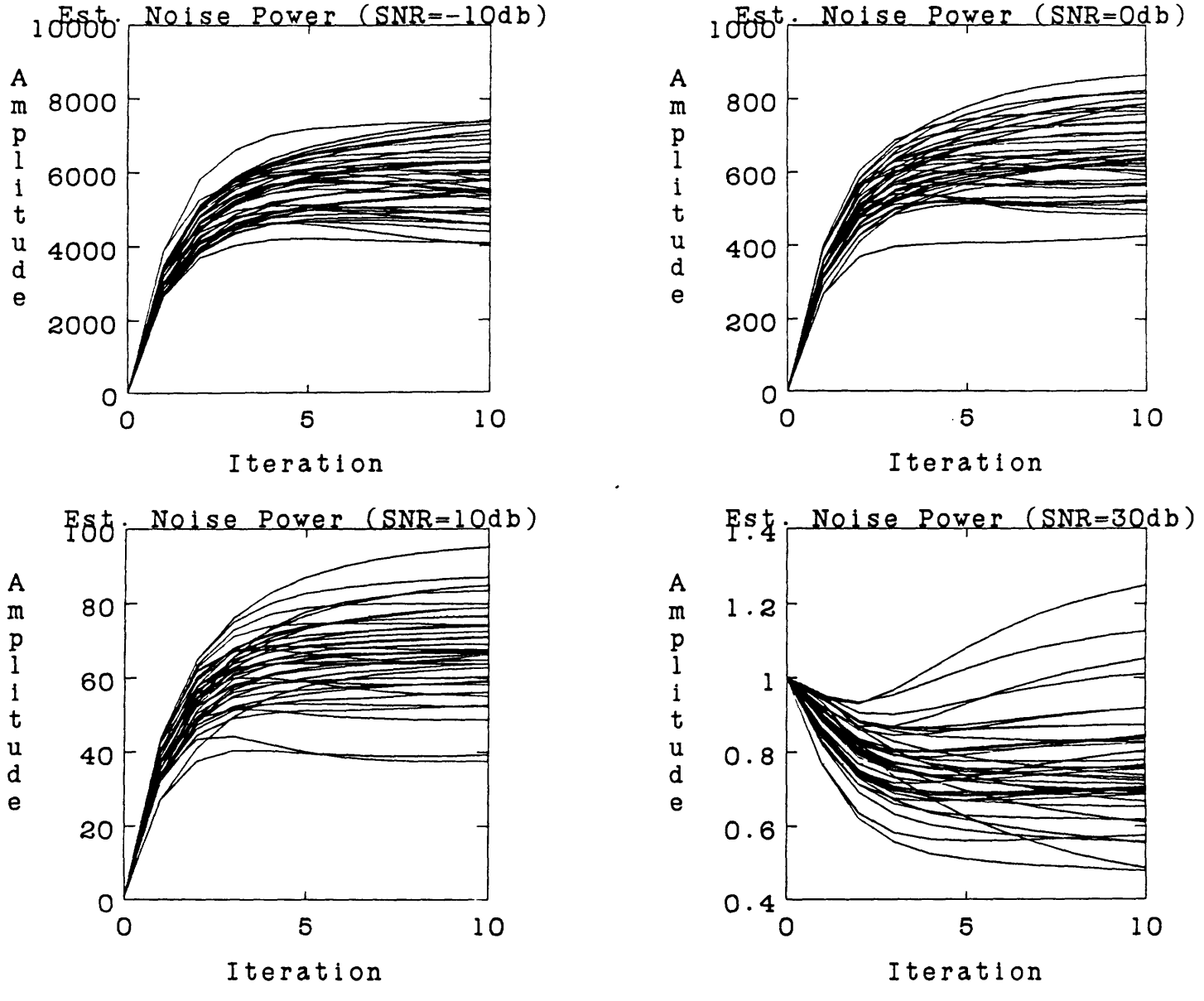


Figure 10f - EM-ML Algorithm, Noise Level Estimates $\hat{\underline{\sigma}}^{(l)}$

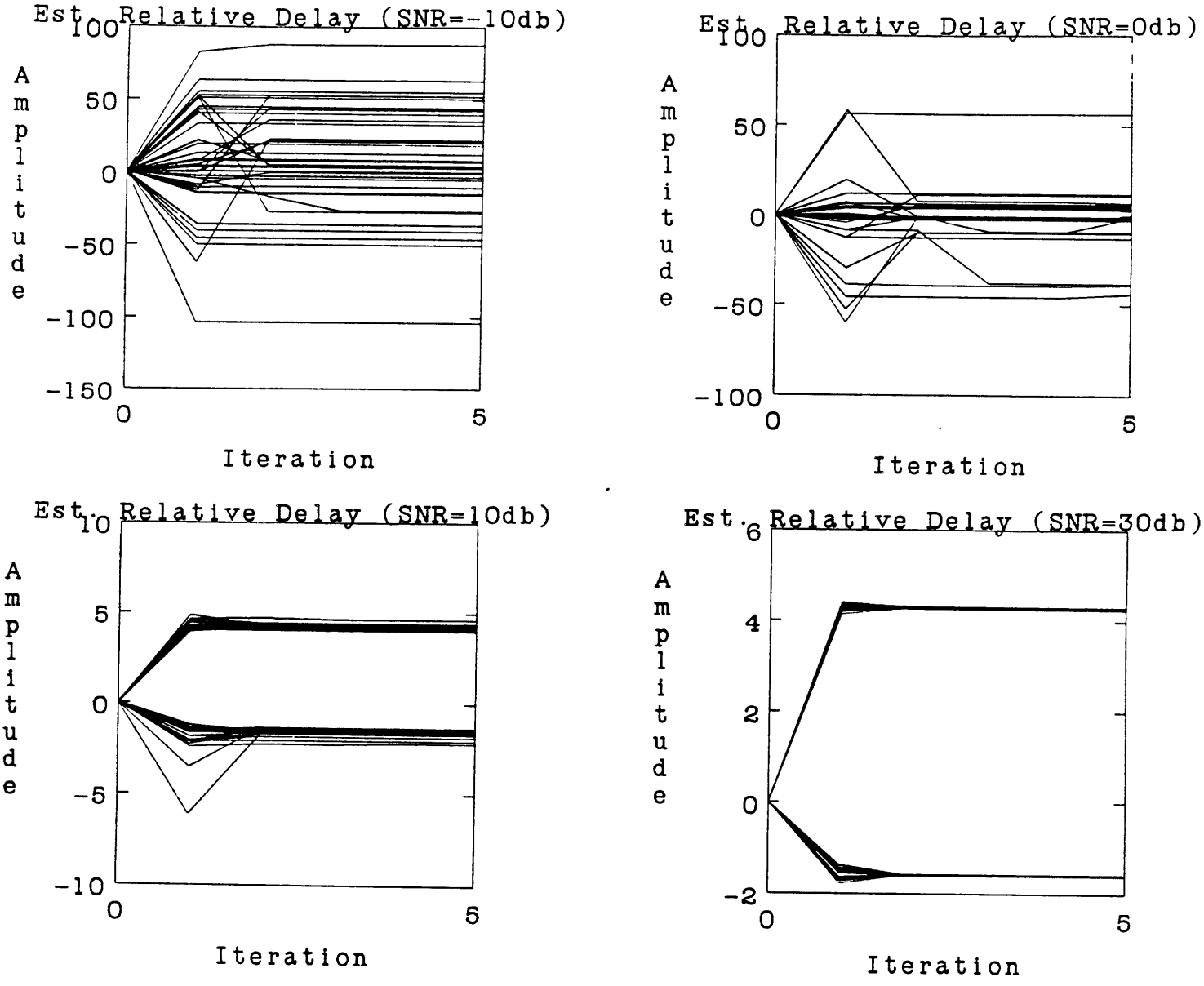


Figure 11a - EM-ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 3$ channels, $\underline{\bar{\tau}} = (-1.58 \;\; 4.29 \;\; 0)^T$, $\underline{\bar{\alpha}} = (3 \;\; 4.5 \;\; 6)^T$, 20 runs for each SNR, estimate 3 delays, 3 gains, 6 pole signal power spectrum with gain, 3 noise level parameters.

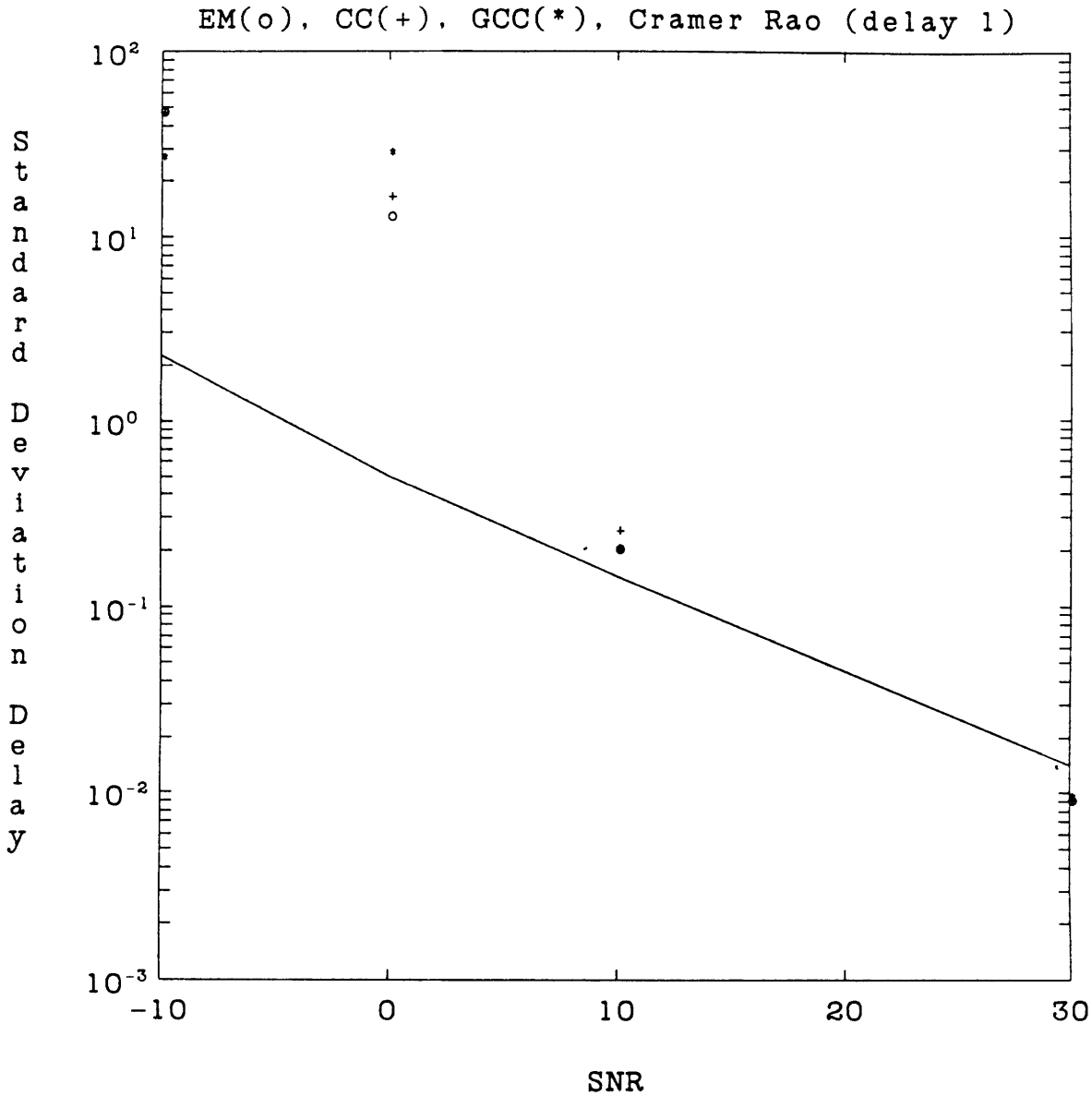


Figure 11b - EM-ML Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_1^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

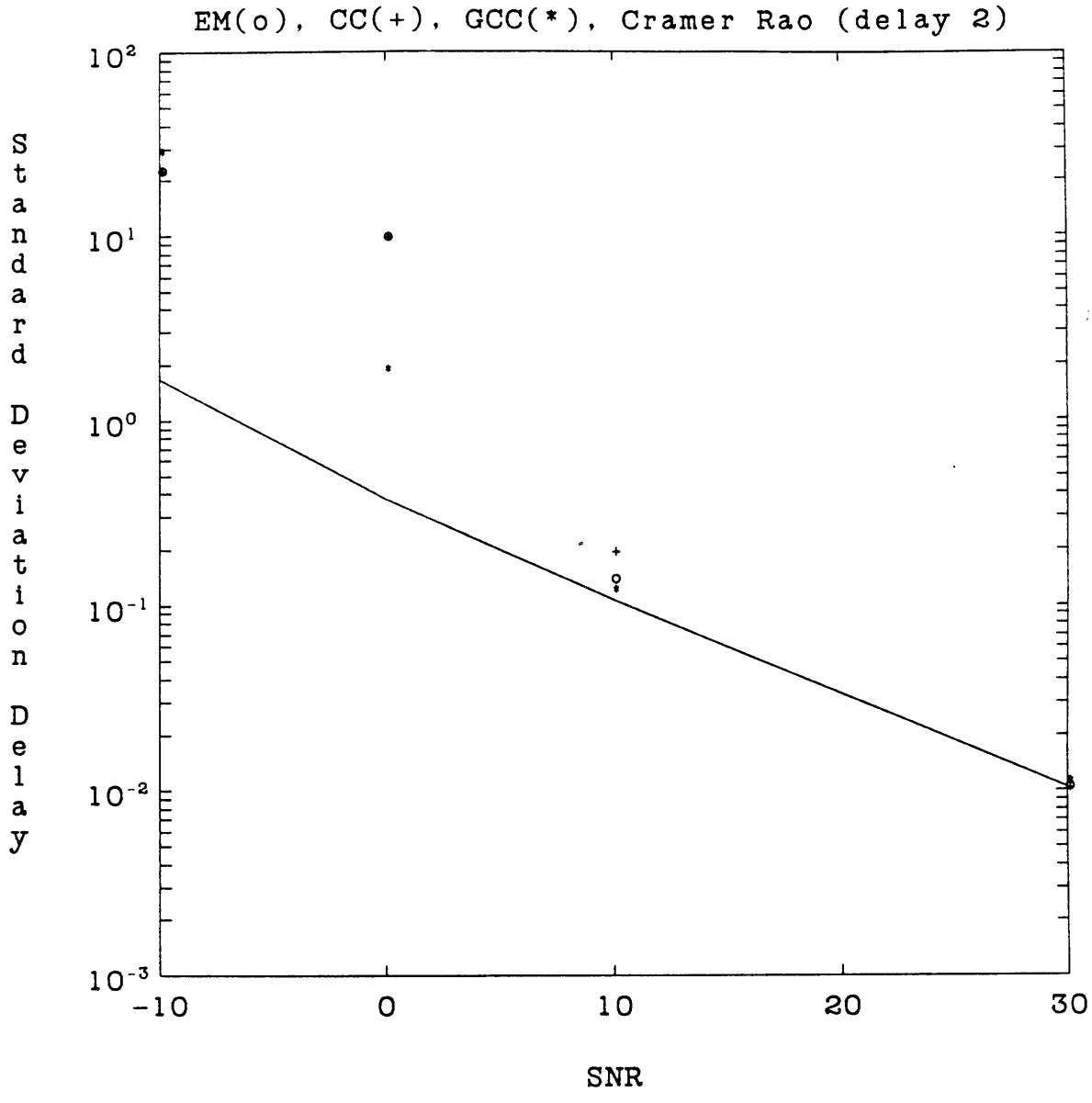


Figure 11c - EM-ML Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

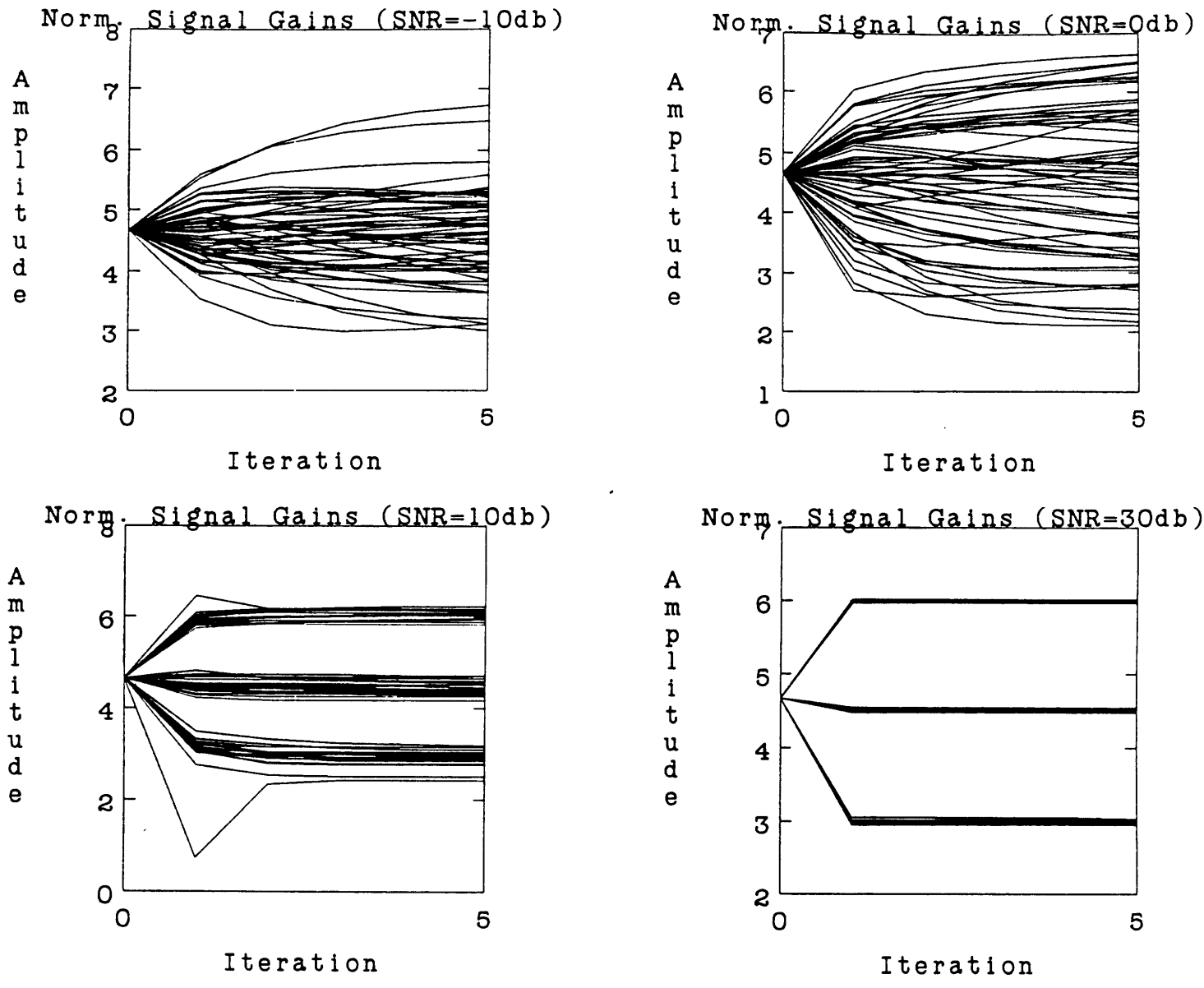


Figure 11d - EM-ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

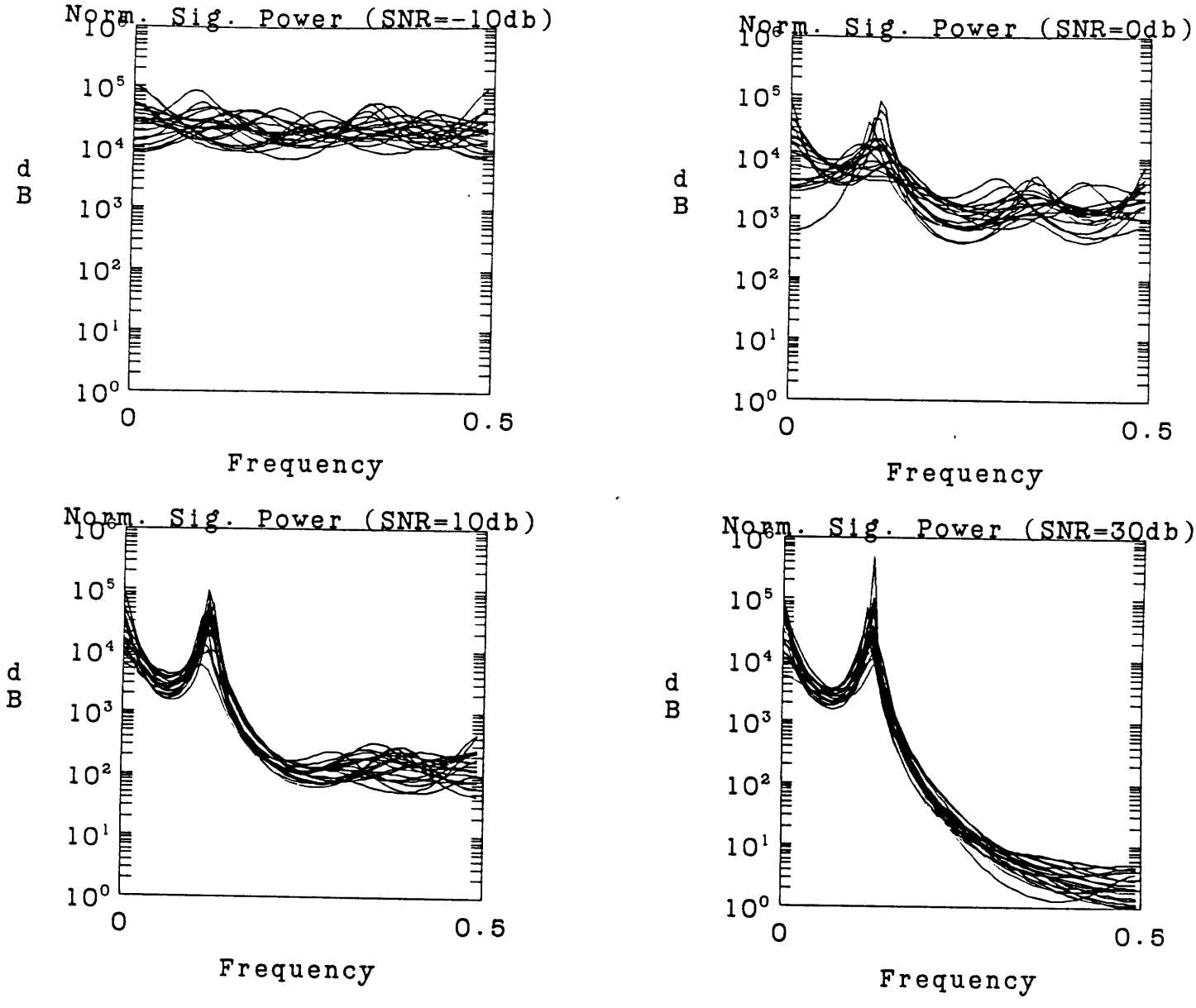


Figure 11e - EM-ML Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

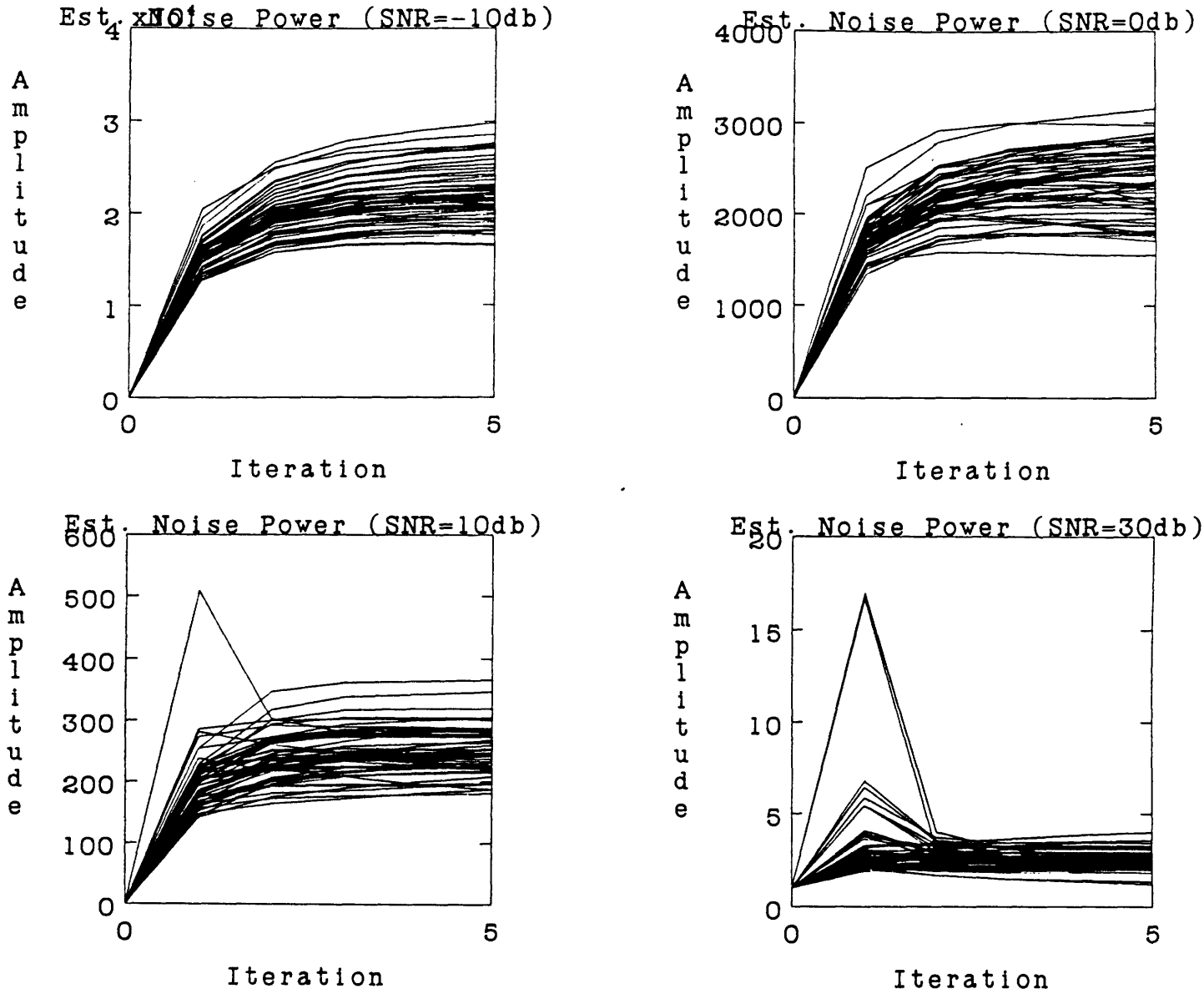


Figure 11f - EM-ML Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

Figure 11 shows the behavior for our $M = 3$ channels example, where all the parameters are estimated, including a 6 pole signal power spectrum model. Above the SNR threshold, the delay estimates converge very rapidly to their final values (within 2 iterations). The convergence rate appears independent of SNR. The standard deviation of the final delay estimates is comparable to the GCC method, and better than CC. The signal gains do not converge to the correct values, but the normalized signal gains do converge quickly for SNR above threshold. The normalized signal power spectrum estimates are accurate near the peaks where the signal is well above the noise floor, although poor in the valleys near the noise floor. Finally the noise spectra level estimates converge rapidly to their correct values. Faster convergence occurs than in the EM algorithm due to the faster convergence of the delay estimates.

Figure 12 shows the behavior for our $M = 8$ channels example, where all the parameters are estimated, including a 10 pole signal power spectrum model. Above the SNR threshold, the delay estimates converge very rapidly to their final values (within 2 iterations). Note, however, that the EM-ML algorithm actually converges no faster than EM in this case. The reason is that EM converges especially quickly when M is large and when the noise level is small. With $M = 8$ peaks in the functions being maximized for each delay, the extra peak that EM has at the old delay estimate does not greatly influence the next delay estimate. The convergence rate of EM-ML, on the other hand, is independent of M or SNR. The standard deviation of the final EM-ML delay estimates is comparable to the GCC method, and better than CC. The signal gains do not converge to the correct values, but the normalized signal gains do converge for SNR above threshold. The convergence rate is comparable to EM. The normalized signal power spectrum estimates are accurate near the peaks where the signal is well above the noise floor, although poor in the valleys near the noise floor. Finally the noise spectra level estimates converge rapidly to their correct values. Faster convergence occurs than in the EM algorithm due to the faster convergence of the delay estimates.

Finally, figure 13 shows the behavior for our flat (white) signal and noise example with $M = 2$ channels. We only show the convergence of the EM-ML delay estimates.

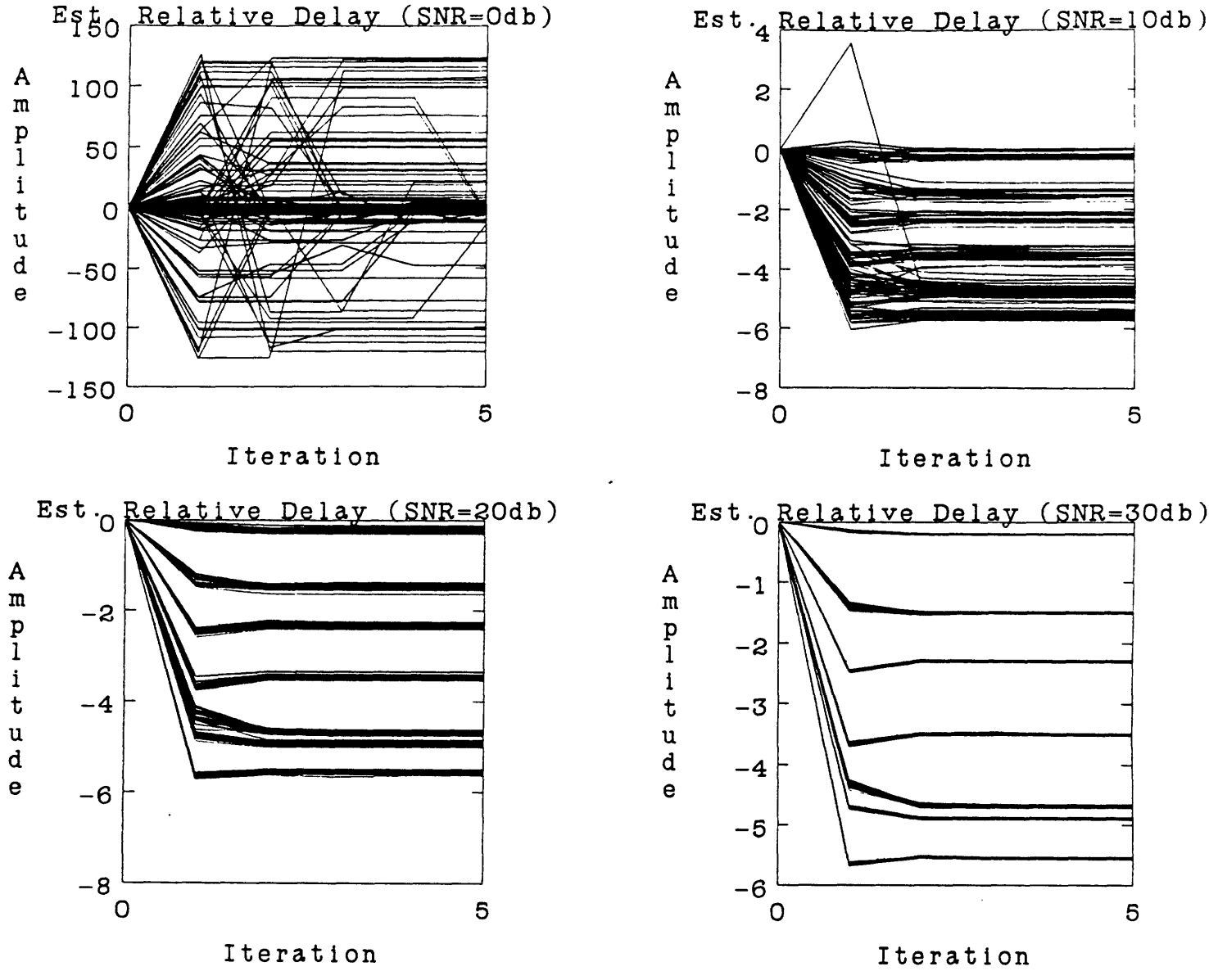


Figure 12a - EM-ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 256$ point data, $M = 8$ channels, $\underline{\tau} = (\,$-4.7 -4.9 -1.5 -3.5 -5.55 -.2 -2.3 0 $)^T$, $\underline{\alpha} = (2\ 3\ 3\ 3\ 5\ 5\ 5\ 5)^T$, 20 runs for each SNR, estimate 8 delays, 8 gains, 10 pole signal power spectrum with gain, 8 noise level parameters.

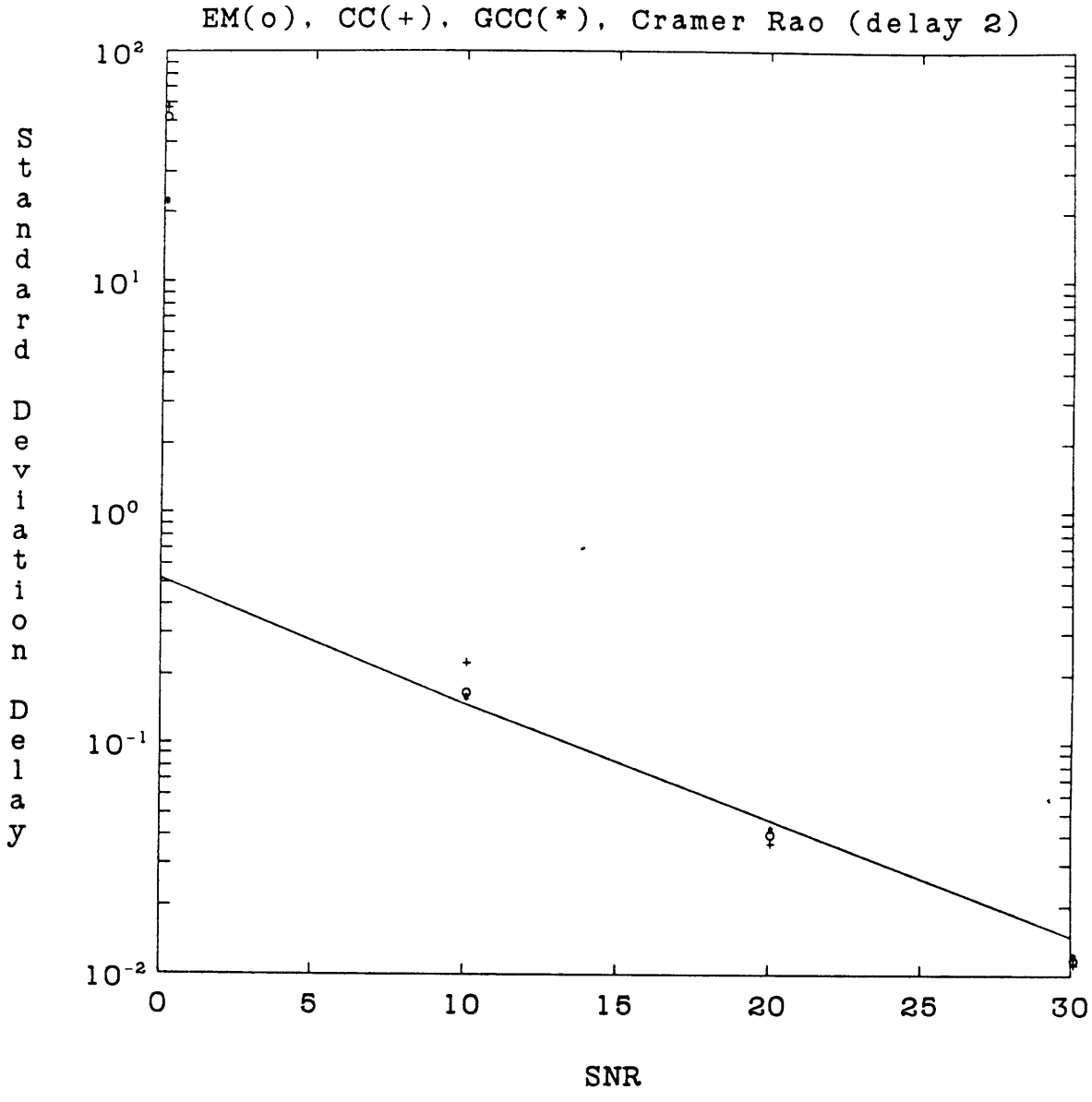


Figure 12b - EM-ML Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_8^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

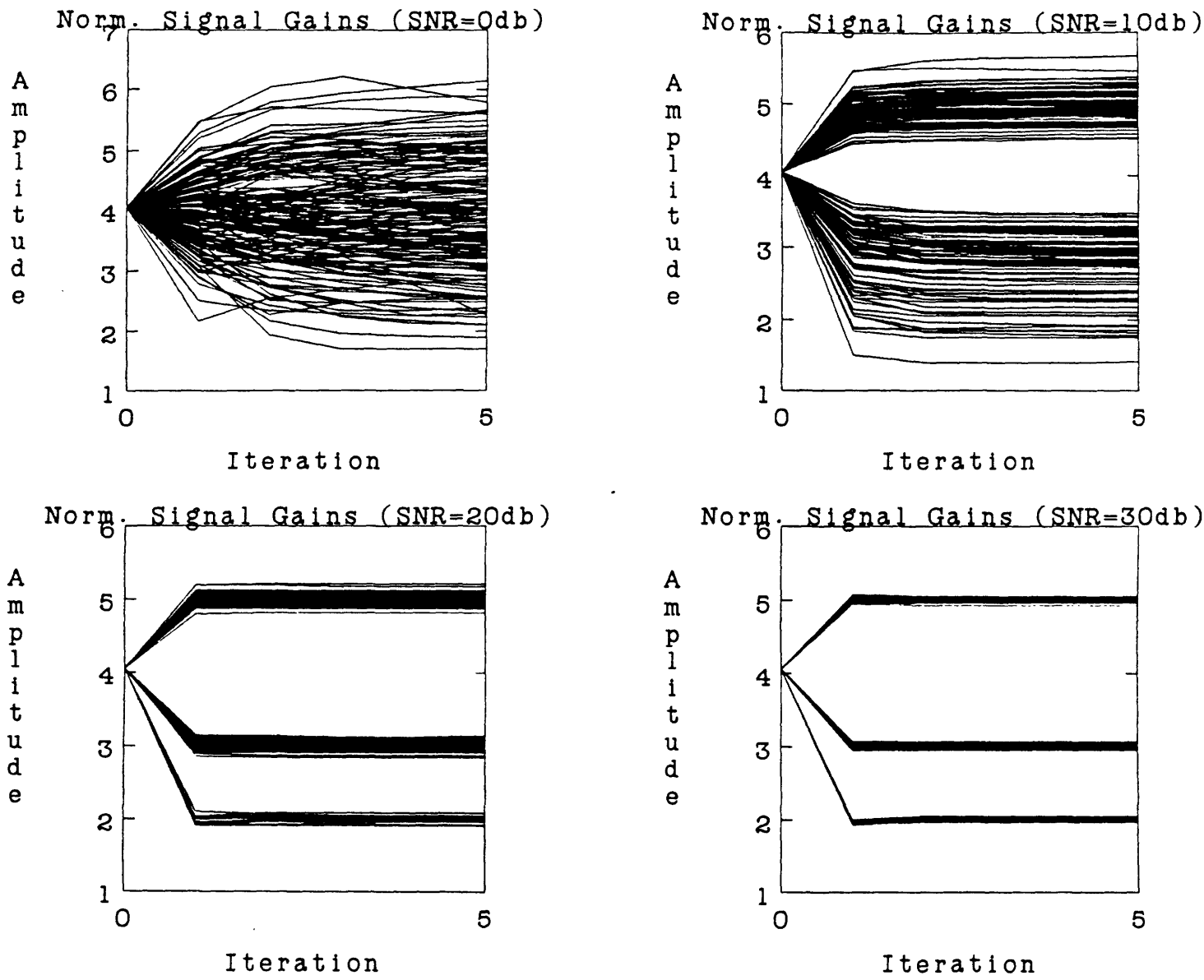


Figure 12c - EM-ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

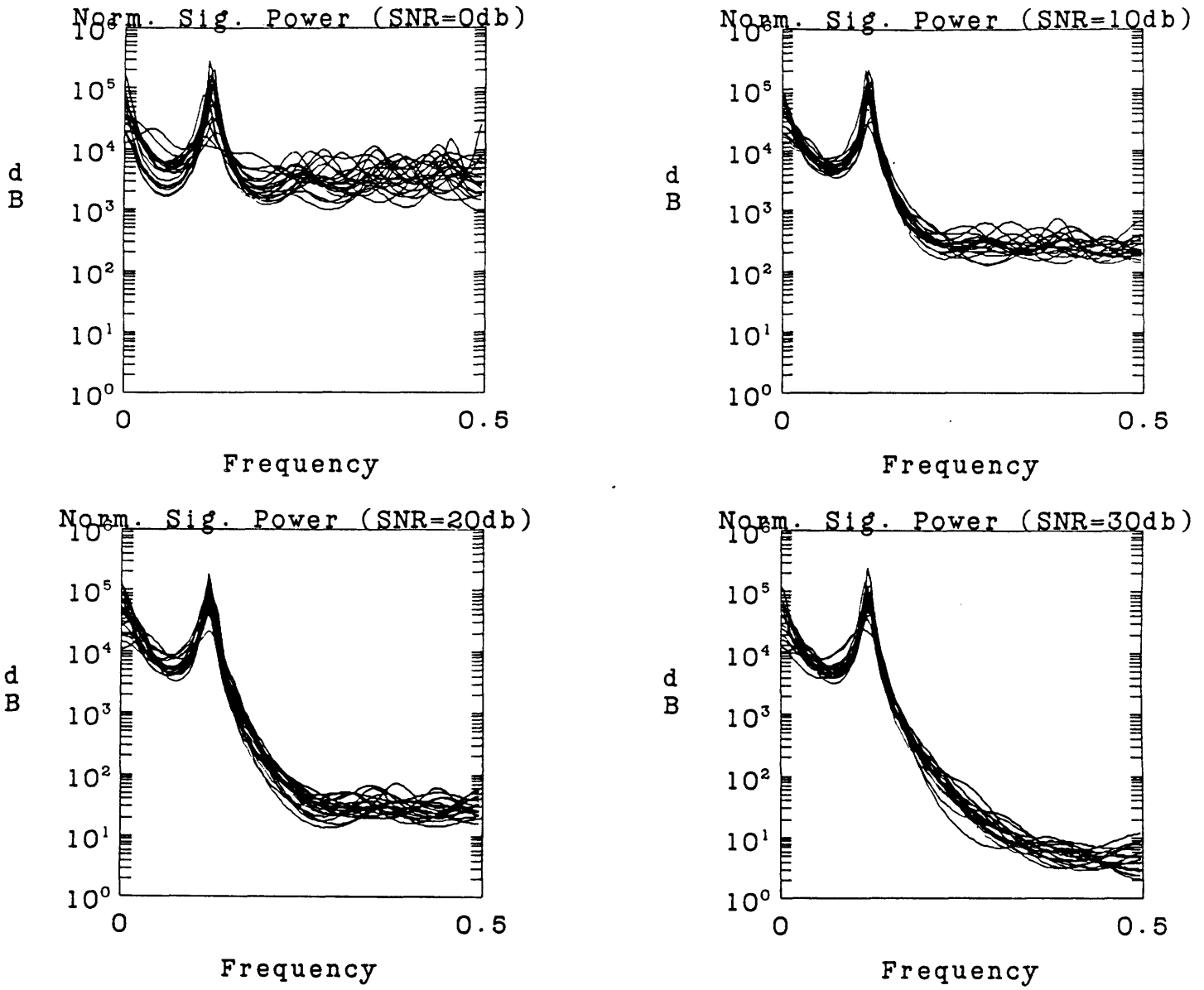


Figure 12d - EM-ML Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

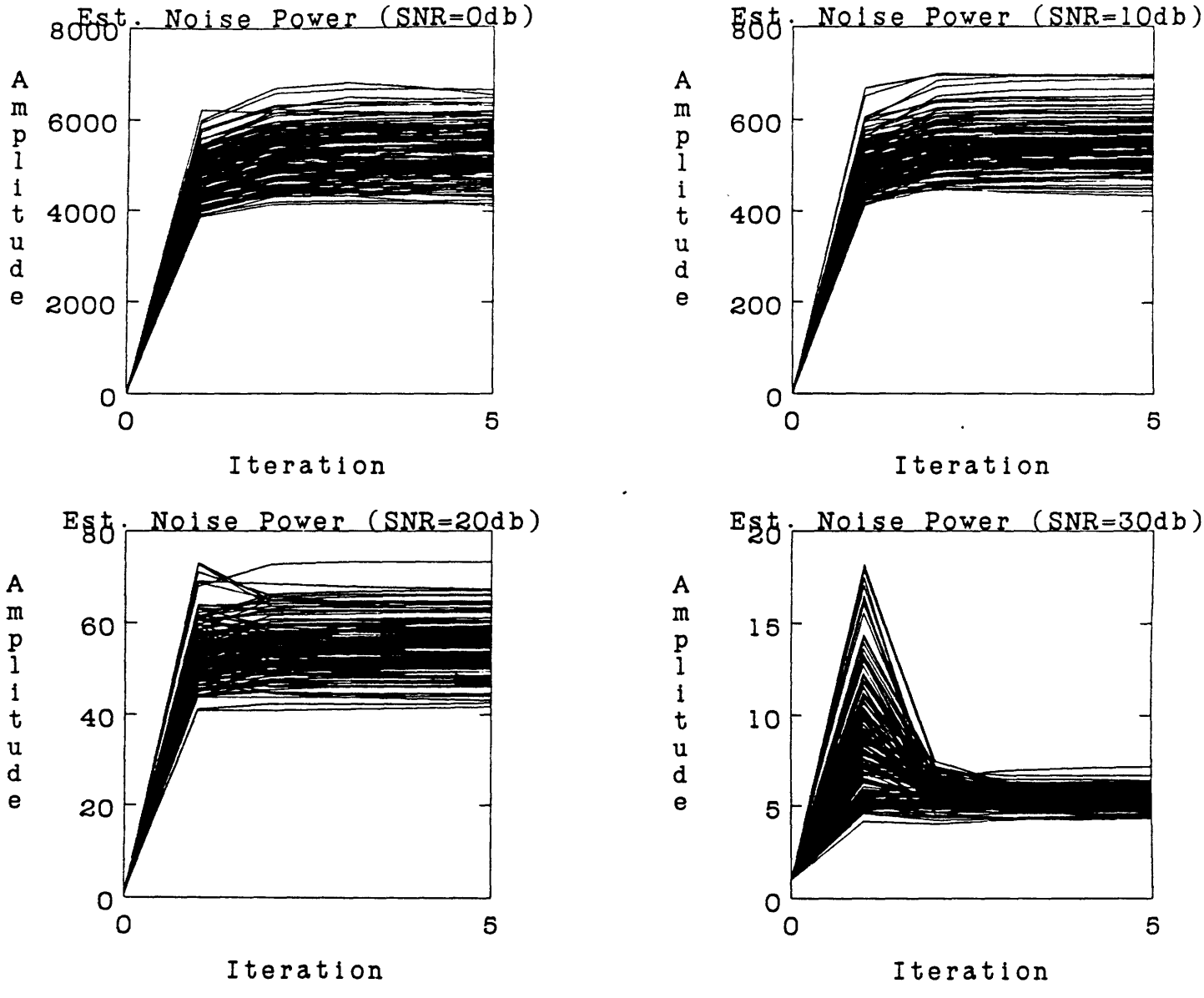


Figure 12e - EM-ML Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

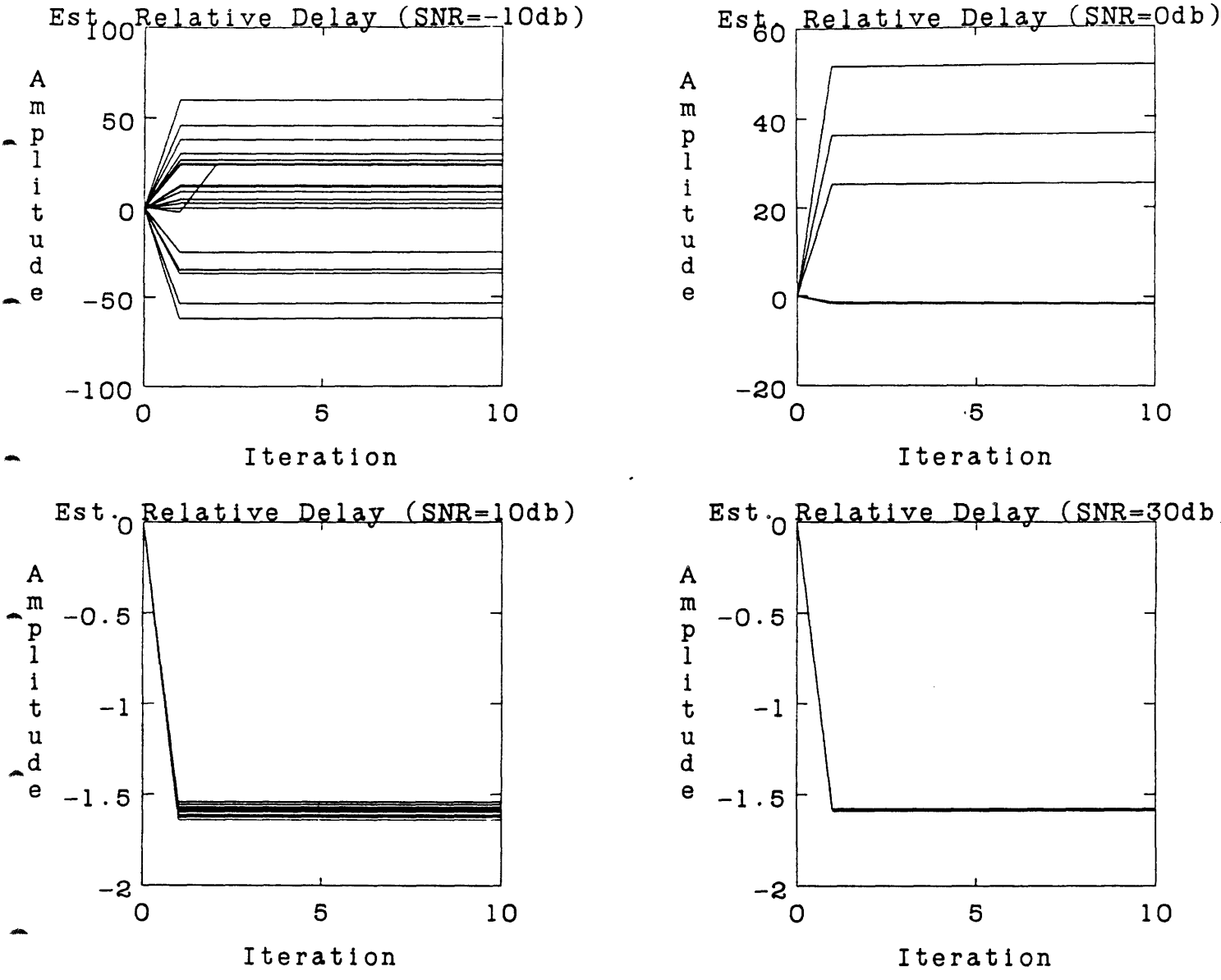


Figure 13a - EM-ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known flat (white) signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains.

Note that for SNR above the threshold, the EM-ML algorithm exhibits none of the confusion shown by EM in this case in figure 8.

7.4 EM-ML Delay/Gain Estimation Algorithm

We ran the same simulations to test the EM-ML delay and gain estimation algorithm. The M delays are estimated one at a time by maximizing the likelihood function over each delay in turn. The M signal gains and a single noise level for all channels are estimated by ML also, which requires solving an eigenvector problem. For convenience in plotting the gain estimates against the actual gains, we choose the normalization constant γ so that the average estimated gains will have the same value as the average actual gains:

$$\sum_{i=1}^{M} \frac{\hat{\alpha}_i^{(l)^2}}{\hat{\sigma}^{(l)}} = \sum_{i=1}^{M} \frac{\bar{\alpha}_i^2}{\bar{\sigma}} = \gamma \tag{137}$$

An EM step is used only to estimate the signal spectrum.

Figure 14 shows the behavior in the $M = 2$ example when we estimate the two time delays together with both signal gains. We assume that the signal and noise spectra are known, and we use the same initial guesses as before. Compare these figures with the EM algorithm in 3 and EM-ML delay estimation in 9. Unlike the previous algorithms, the EM-ML delay/gain algorithm achieves convergence of both delays and gains in one iteration at all SNR. As predicted, the convergence rate is superlinear. This is a clear improvement over the previous algorithms.

Figure 15 shows the behavior when we estimate all the parameters: two time delays, plus both signal gains, a 6 pole signal spectrum with gain g^2, and the noise level on all receivers. Initial guesses were chosen as before, and we used the same data. Compare with EM in figure 5 and EM-ML Delay estimation in figure 10. The convergence rate and standard deviation of the delays is virtually identical to that in the EM-ML delay estimation algorithm. Unlike the promise of the theory or the previous example, however, convergence of the gains is *not* superlinear. The problem is that the noise gain estimate is set far too high after the first iteration, due to the initial error in the delay estimates. Because the signal gain estimates scale proportionally to the noise

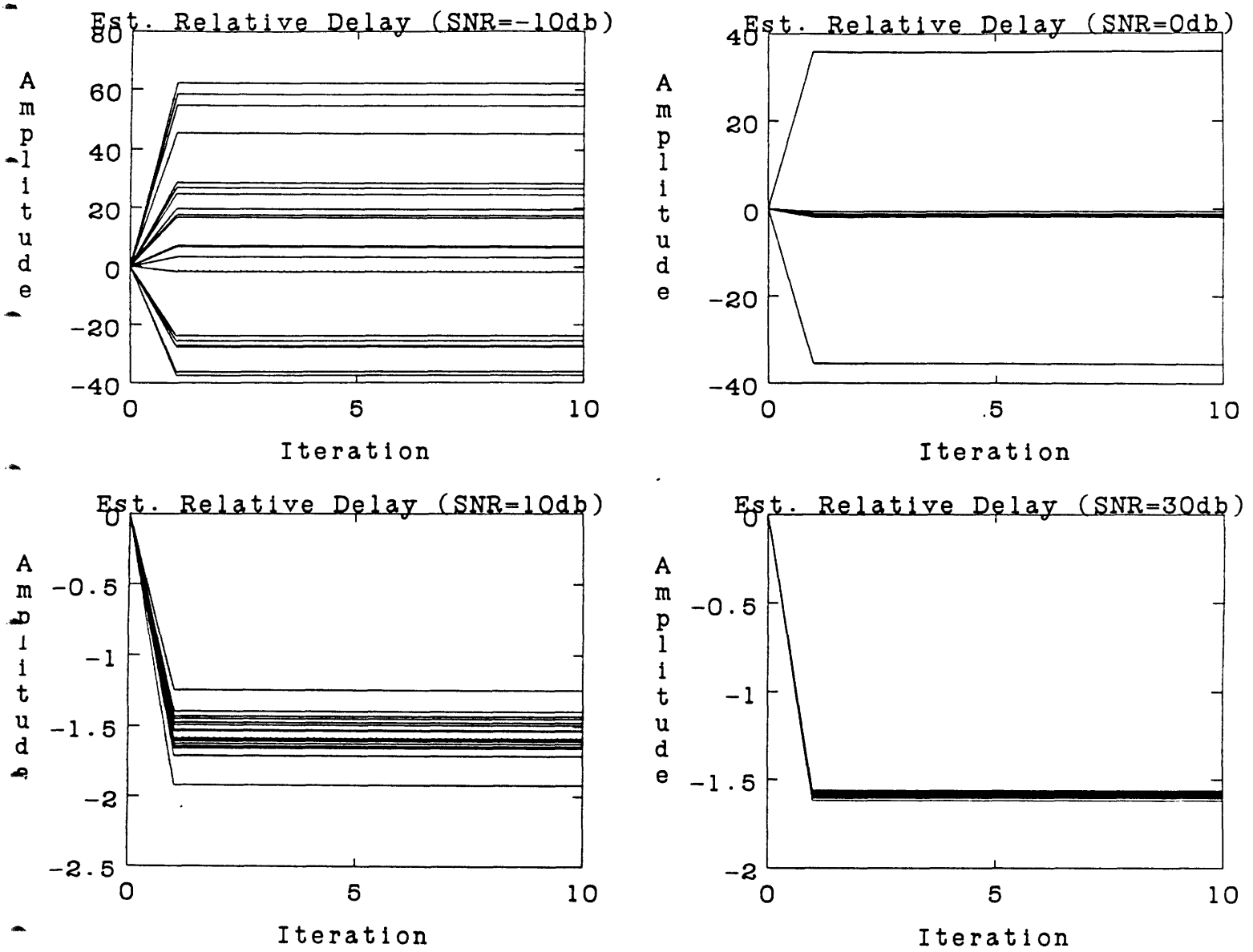


Figure 14a - EM-ML Delay/Gain Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - known lowpass signal model, known noise levels, 128 point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains.

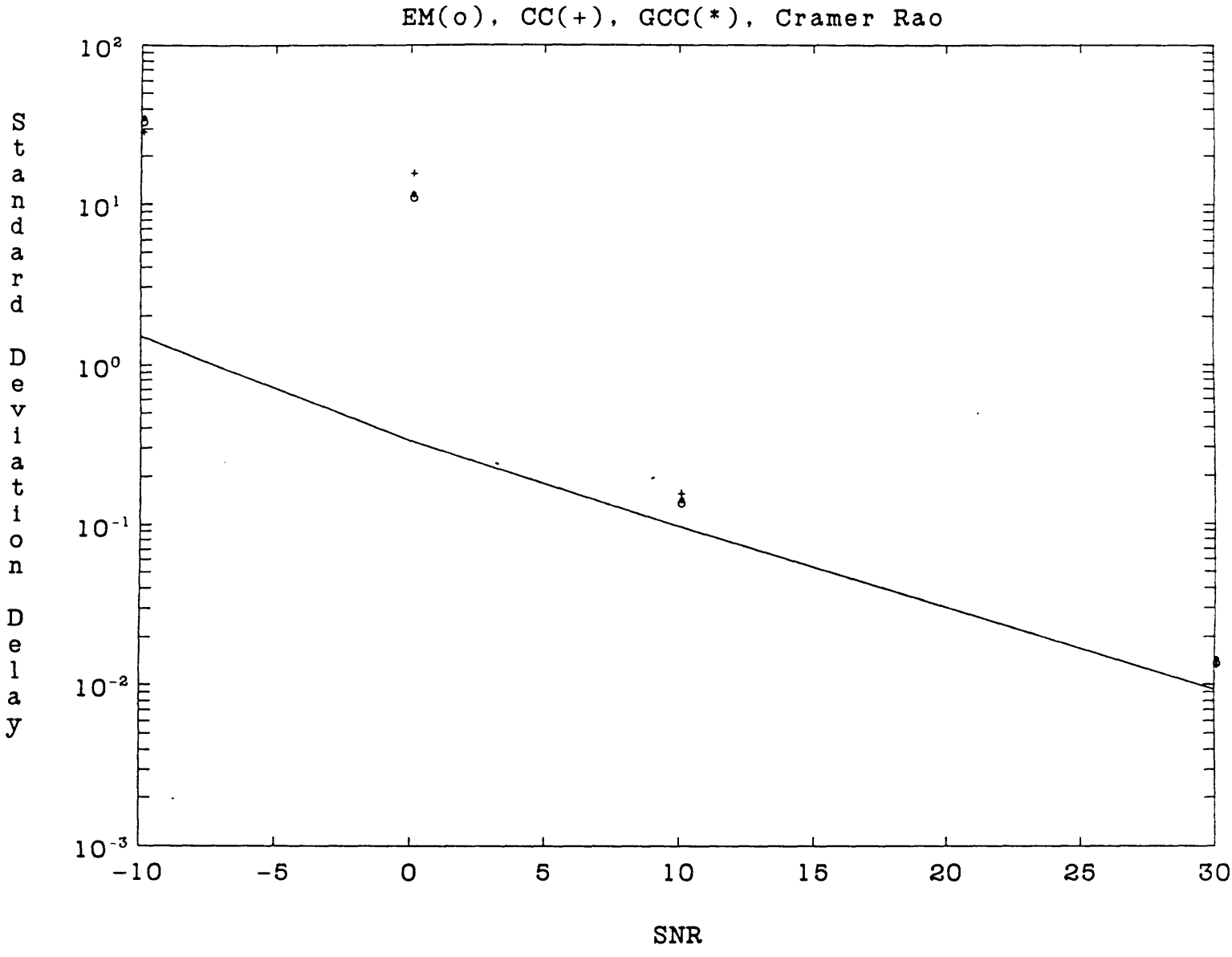


Figure 14b - EM-ML Delay/Gain Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

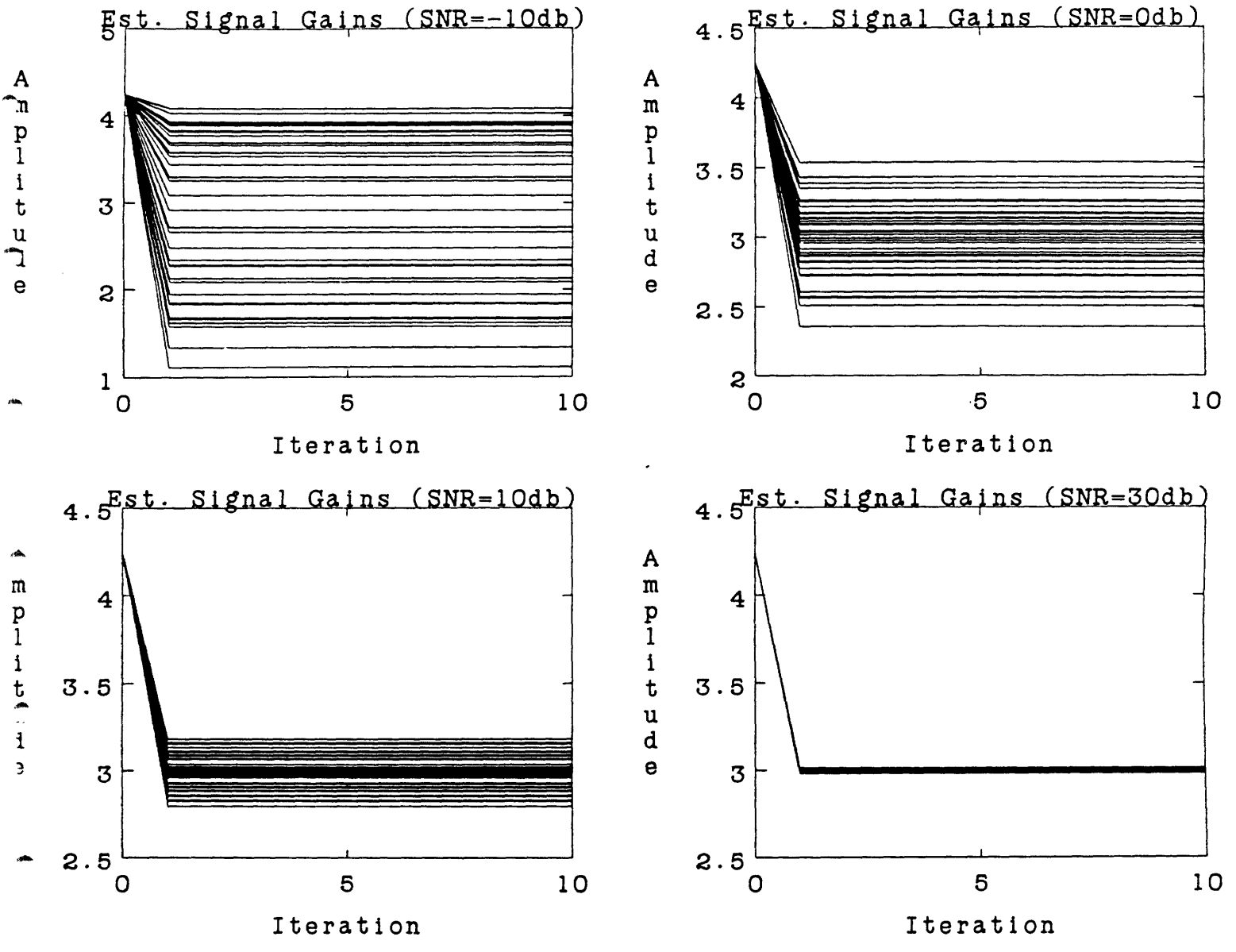
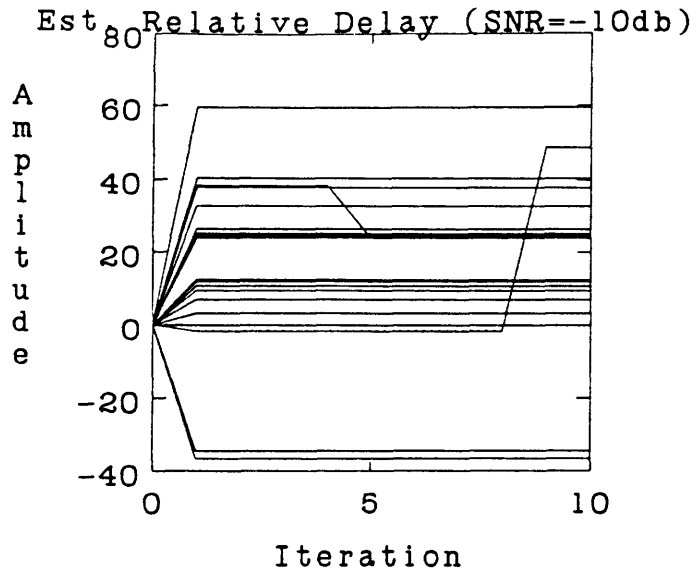
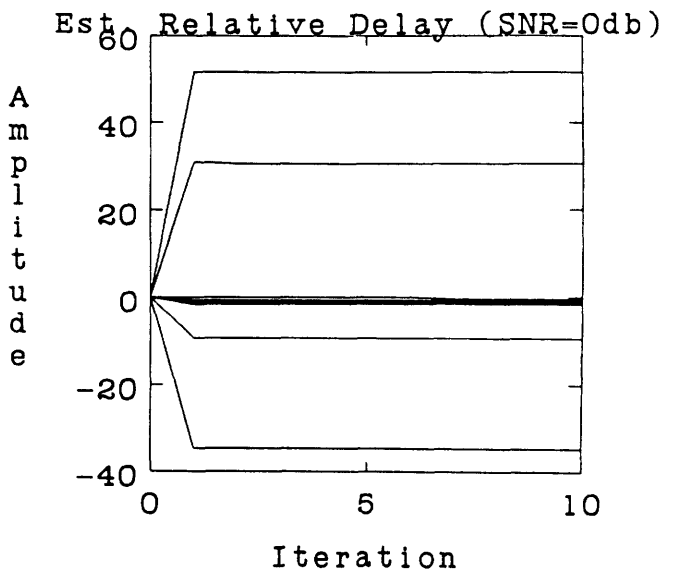
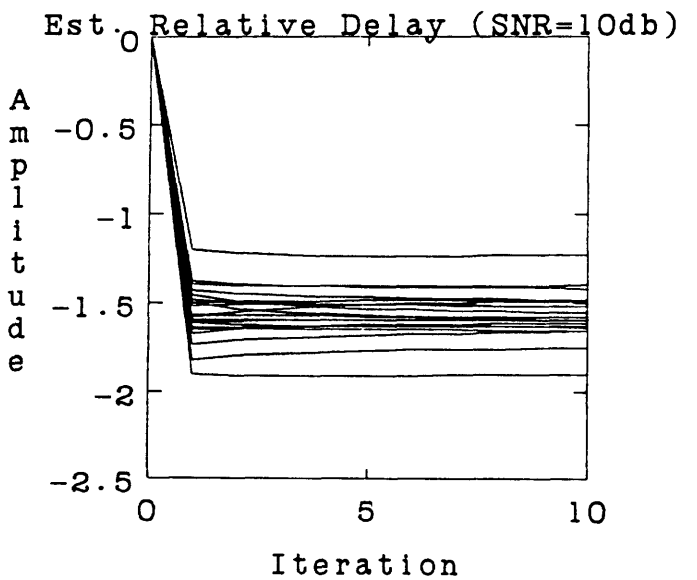


Figure 14c - EM-ML Delay/Gain Algorithm, Signal Gain Estimates $\hat{\underline{\alpha}}^{(l)}$







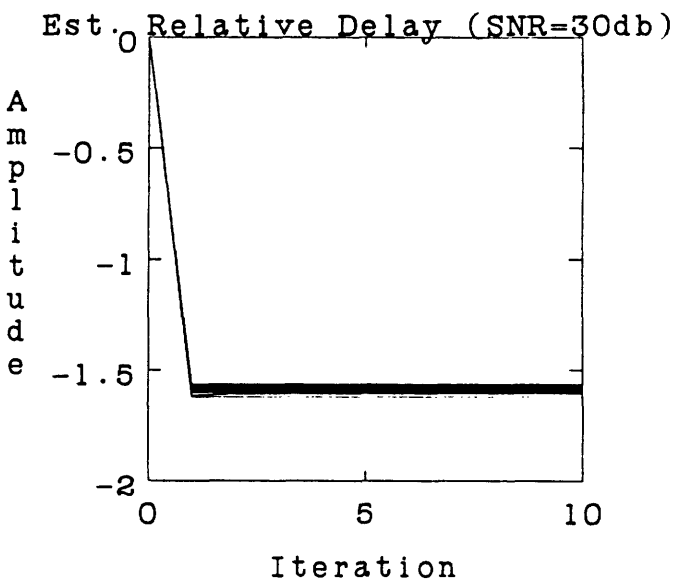


Figure 15a - EM-ML Delay/Gain Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3 \;\; 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains, 6 pole signal power spectrum with gain, 2 noise level parameters.

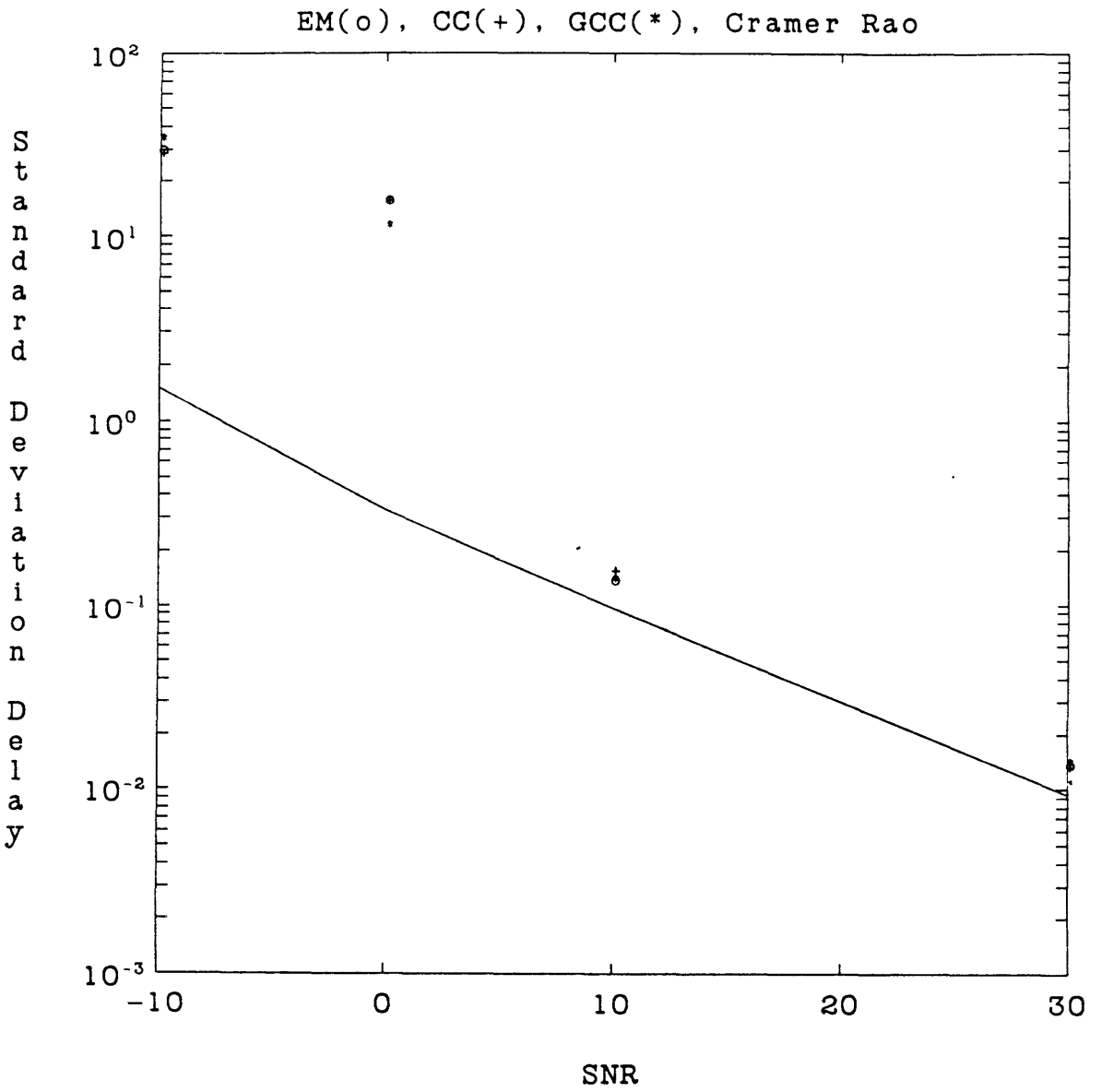


Figure 15b - EM-ML Delay/Gain Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

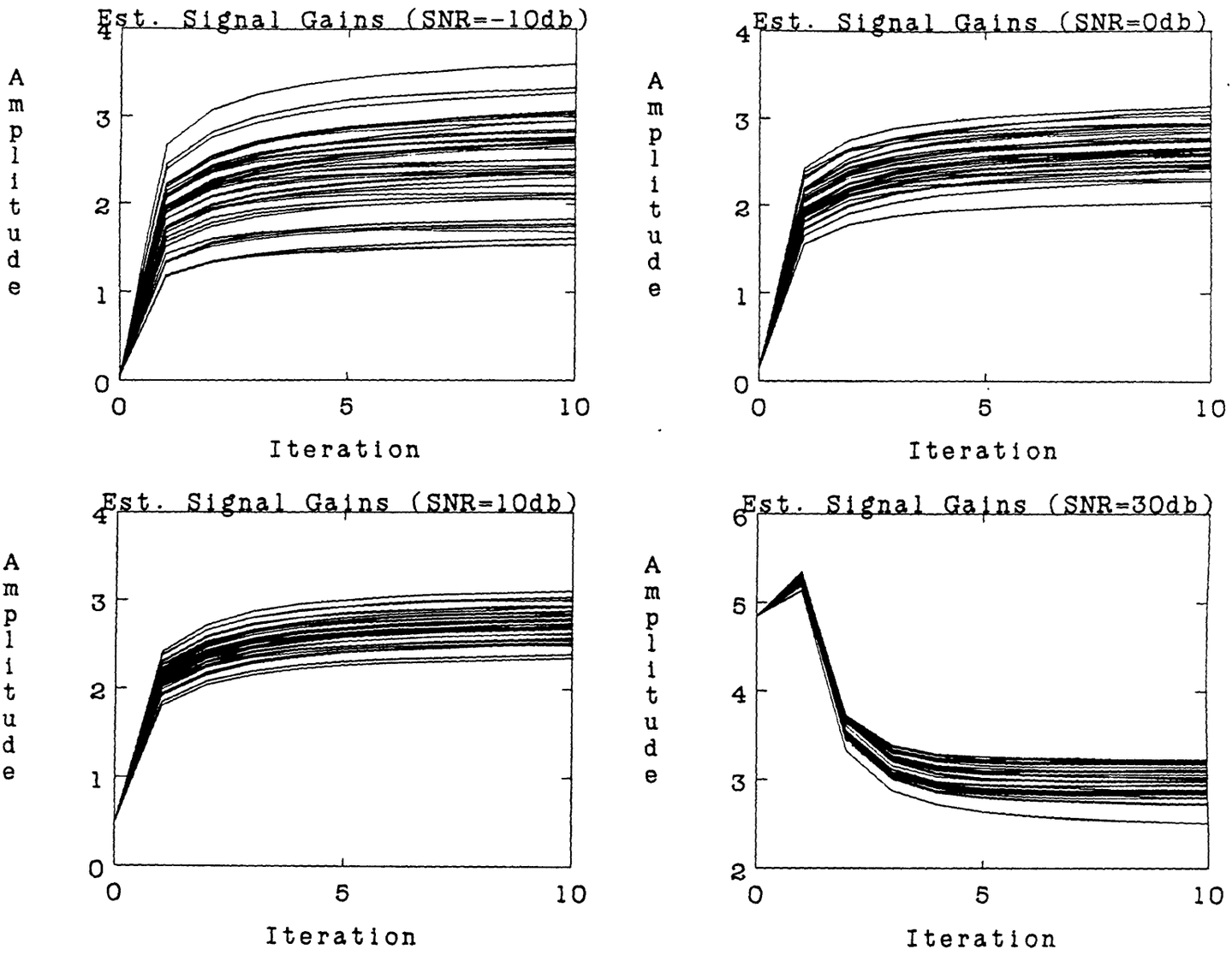


Figure 15c - EM-ML Delay/Gain Algorithm, Signal Gain Estimates $\underline{\hat{\alpha}}^{(l)}$.

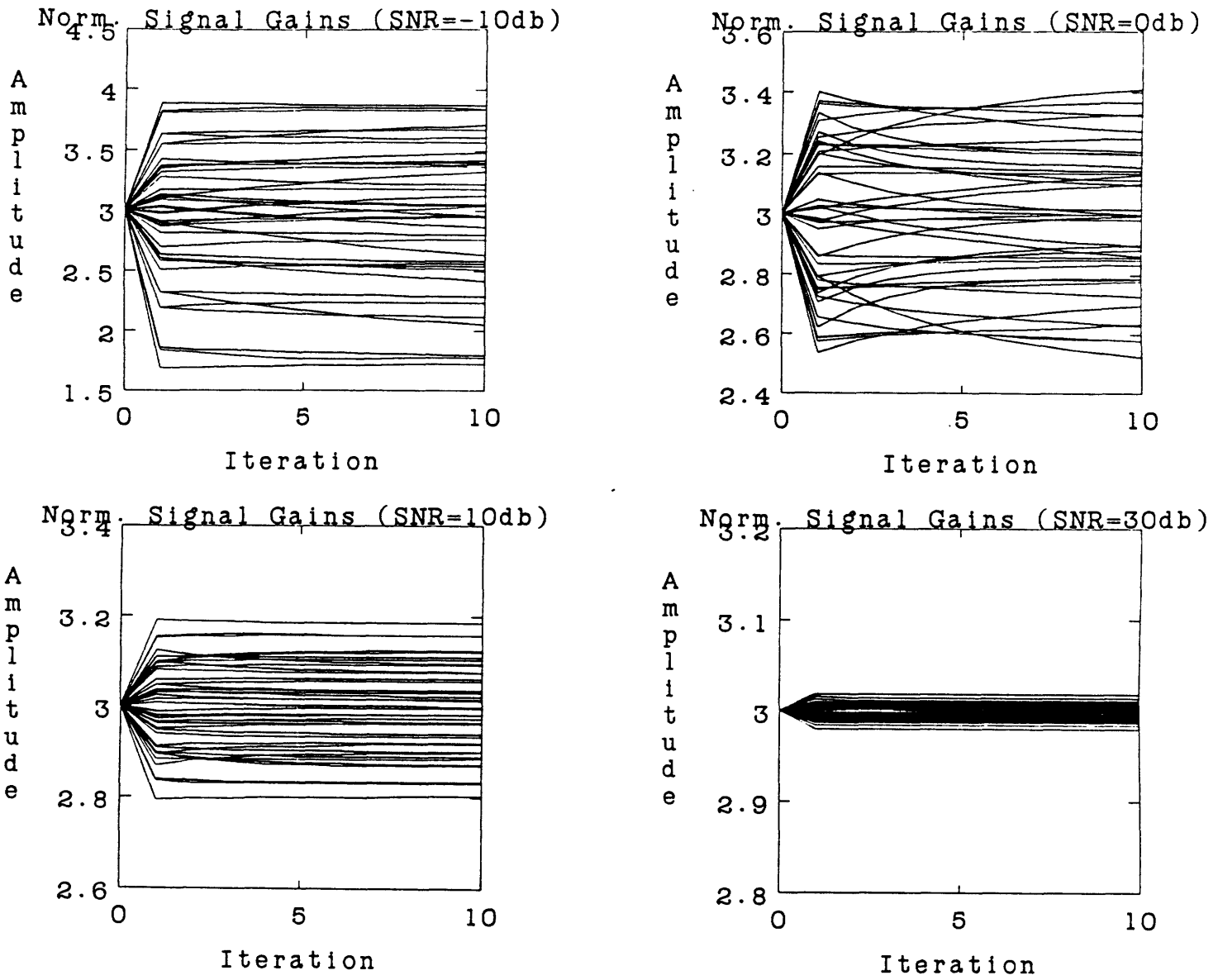


Figure 15d - EM-ML Delay/Gain Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

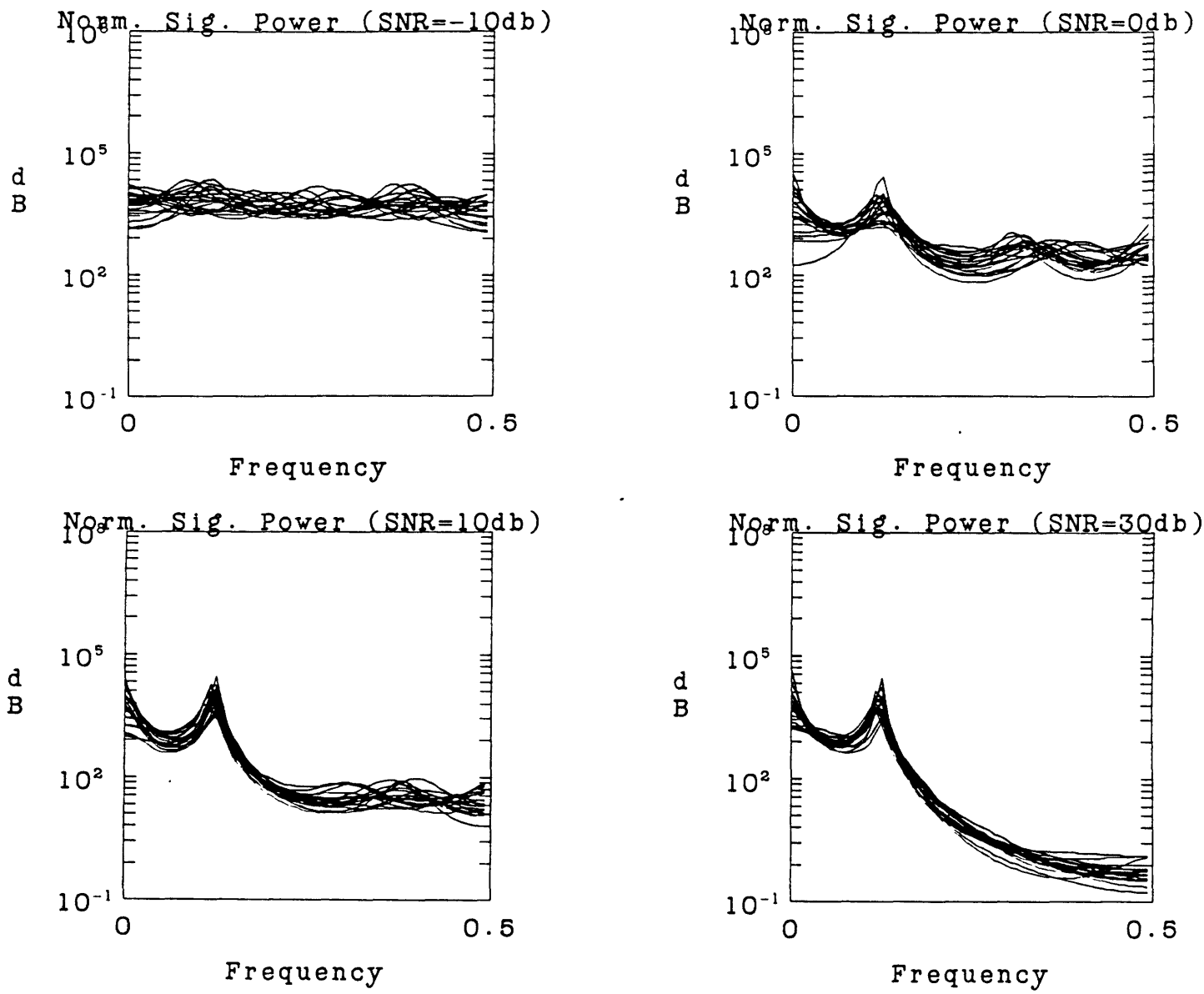


Figure 15e - EM-ML Delay/Gain Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

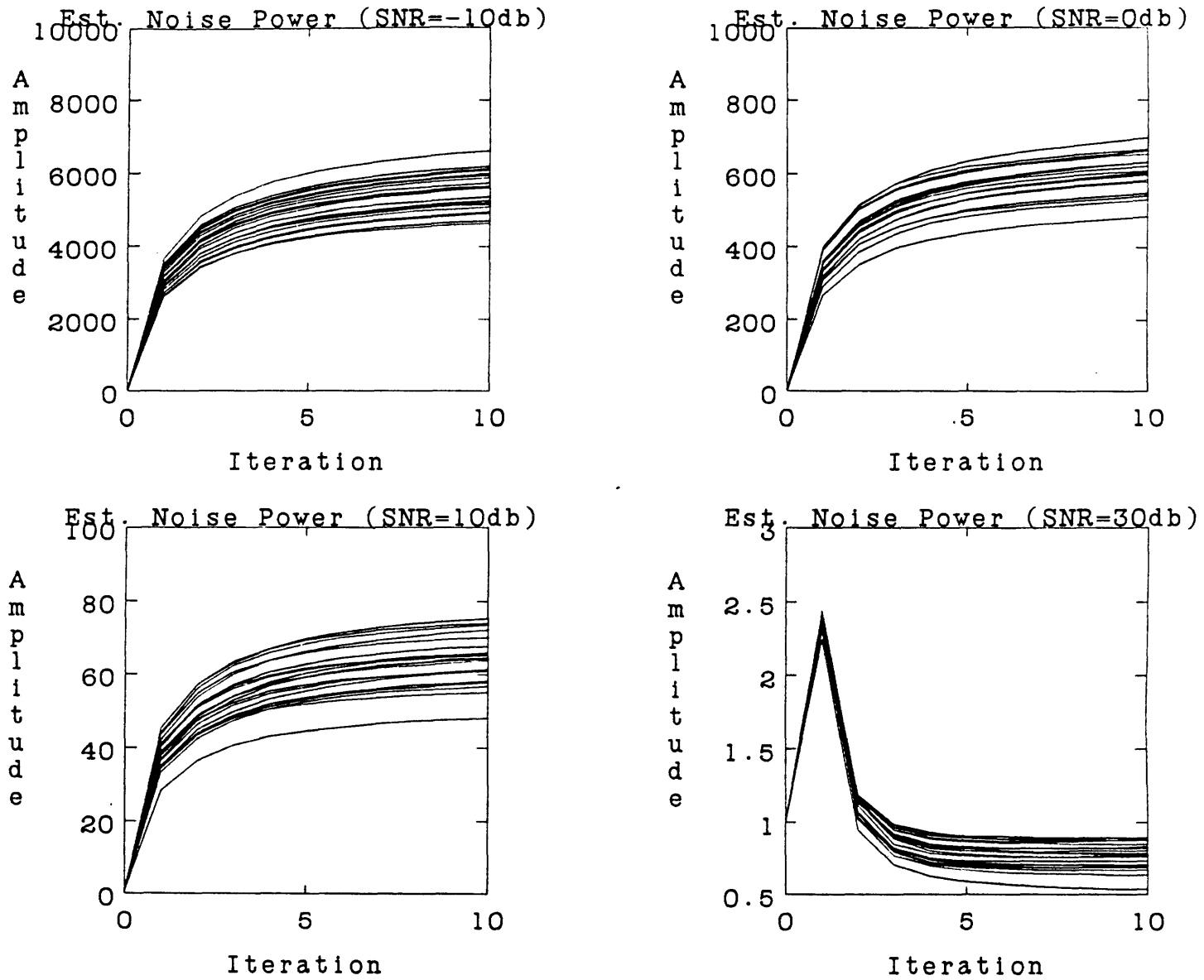


Figure 15f - EM-ML Delay/Gain Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

level, they are also set far too high after the first iteration. If we ignore the overall level of the signal gains, and plot only the normalized gains $\underline{\hat{\alpha}}^{(l)}/\kappa^{(l)}$, as in the previous methods, then convergence of the relative gains is indeed faster than in EM or in the EM-ML delay algorithm. Convergence of the power spectra is similar to the other algorithms. The noise level, however, converges more quickly and with less variance than in the other algorithms. This, however, is because this algorithm only estimates a single noise level for all channels, while the others estimate separate noise levels for each channel.

Figure 16 shows the behavior for our $M = 3$ channels example, where all the parameters are estimated, including a 6 pole signal power spectrum model. Our conclusions for this case are virtually the same as in the previous example. The delay estimates converge in the same manner as in the EM-ML delay estimation algorithm. Near threshold, the normalized gains converge more quickly than in the other algorithms. At SNR=10dB and up, however, the difference is hardly noticeable. The power spectral estimates converge in the same manner as before, as do the noise level estimates, though the variance of the noise level is lower than before due to the use of only a single noise level for all 3 channels.

Figure 17 shows the behavior for our $M = 8$ channels example, where all the parameters are estimated, including a 10 pole signal power spectrum model. The conclusions are nearly the same as in the $M = 3$ case. The delay estimates are virtually unchanged from the EM-ML delay algorithm, the normalized gain estimates converge faster only near threshold SNR, the noise level variance is lower due to the use of a single parameter for all 8 channels, and the signal power spectra converge in the same manner as before.

The advantages of the EM-ML delay/gain estimation algorithm thus only appear striking when the noise level and signal power spectrum is known. In this case, convergence really does appear superlinear. Otherwise, the convergence rate of the signal gains is only significantly better than in the EM-ML delay estimation algorithm when SNR is near threshold. The quality of the delay estimates in unchanged from the EM-ML delay estimation algorithm. The overall advantage of this EM-ML delay/gain

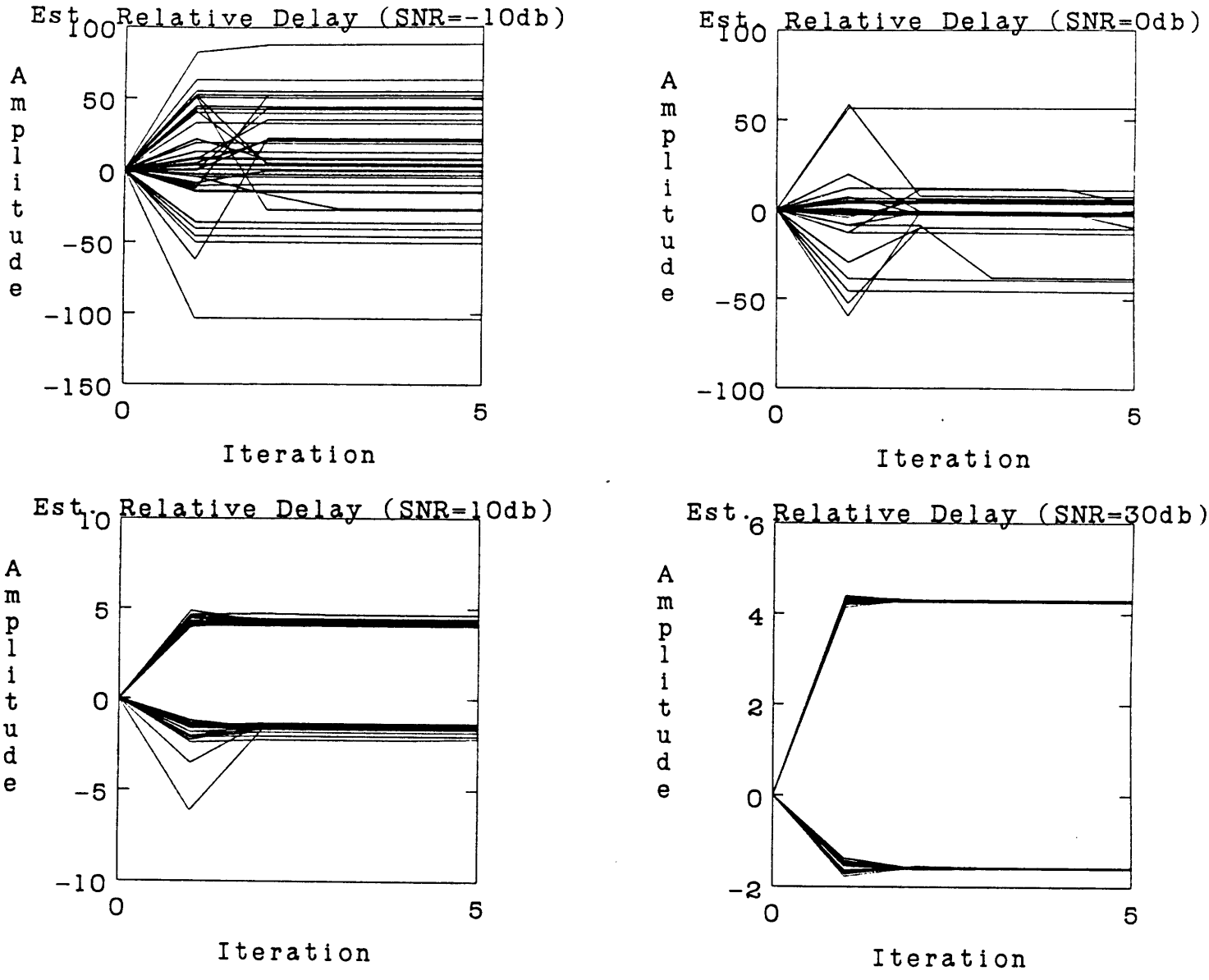


Figure 16a - EM-ML Delay/Gain Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 3$ channels, $\underline{\bar{\tau}} = (-1.58 \;\; 4.29 \;\; 0)^T$, $\underline{\bar{\alpha}} = (3 \;\; 4.5 \;\; 6)^T$, 20 runs for each SNR, estimate 3 delays, 3 gains, 6 pole signal power spectrum with gain, 3 noise level parameters.

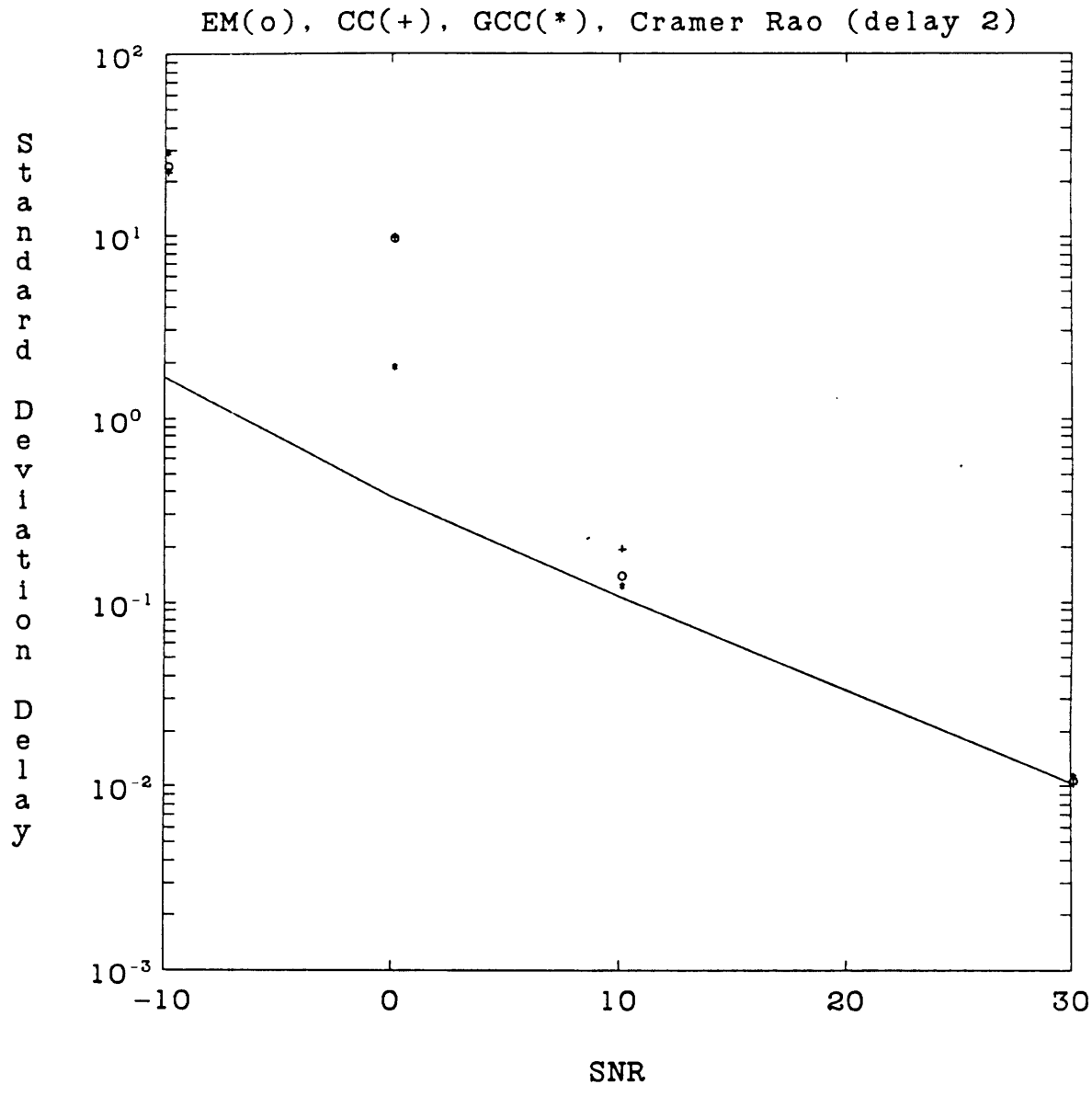


Figure 16b - EM-ML Delay/Gain Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

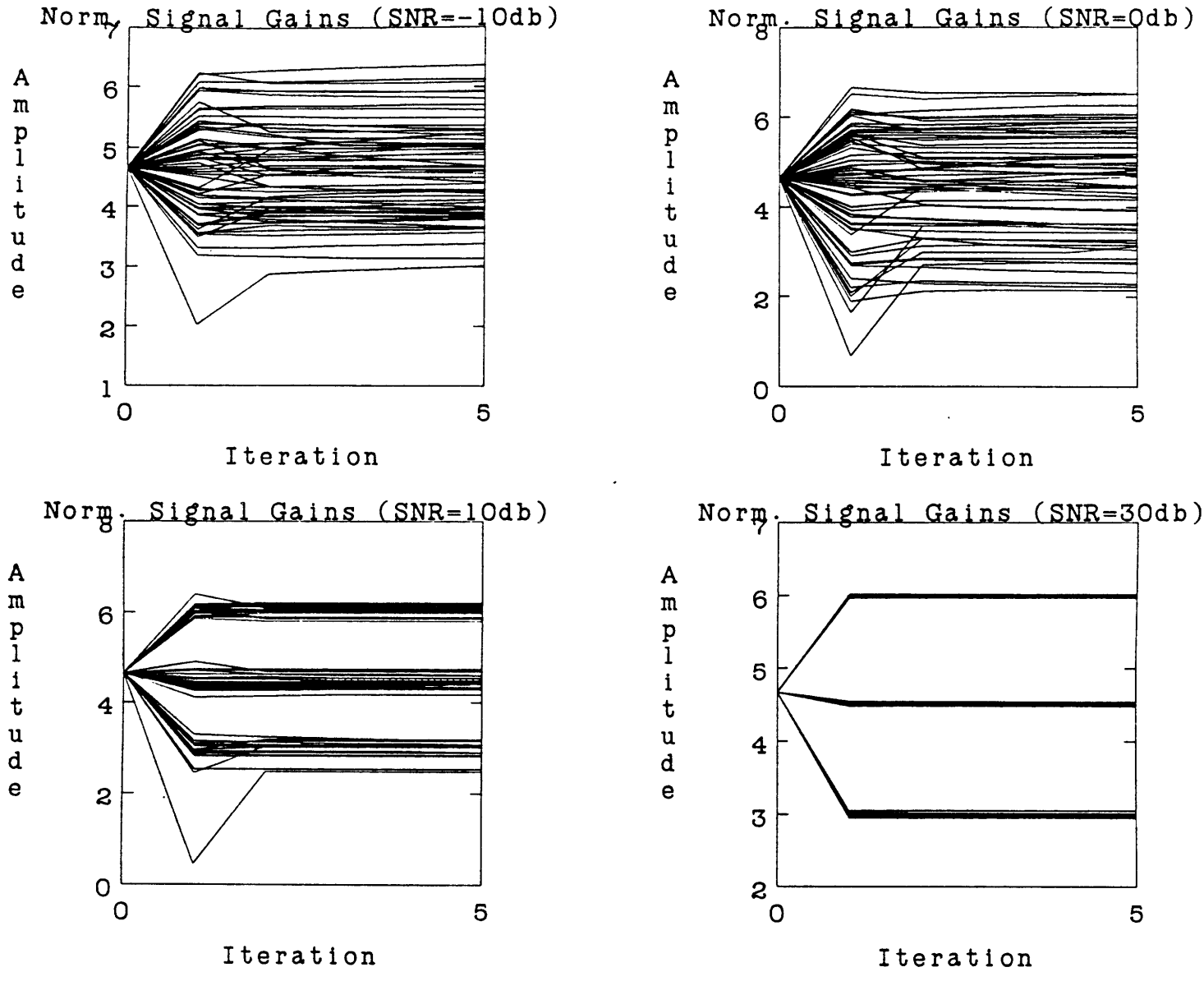


Figure 16c - EM-ML Delay/Gain Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

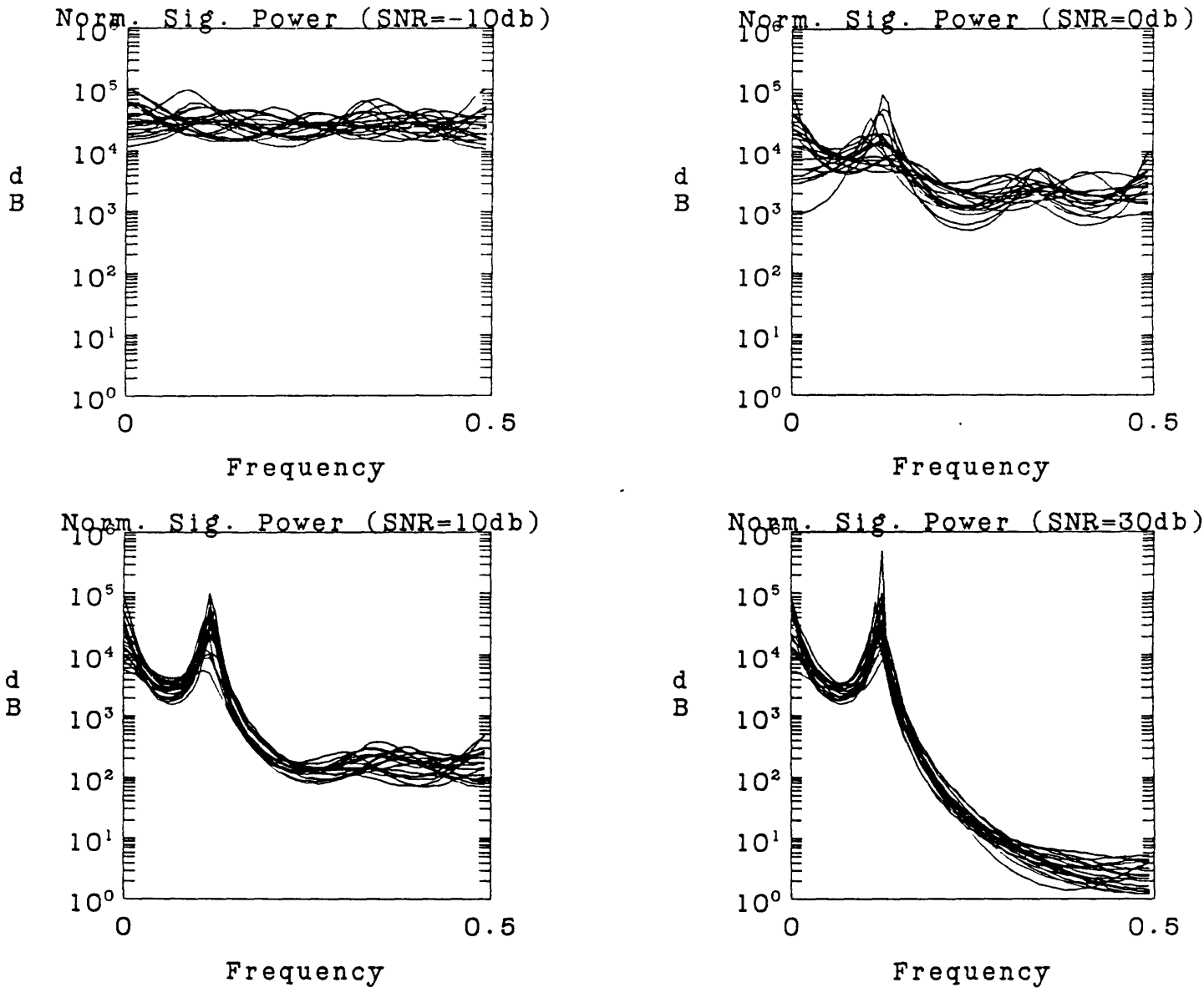


Figure 16d - EM-ML Delay/Gain Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

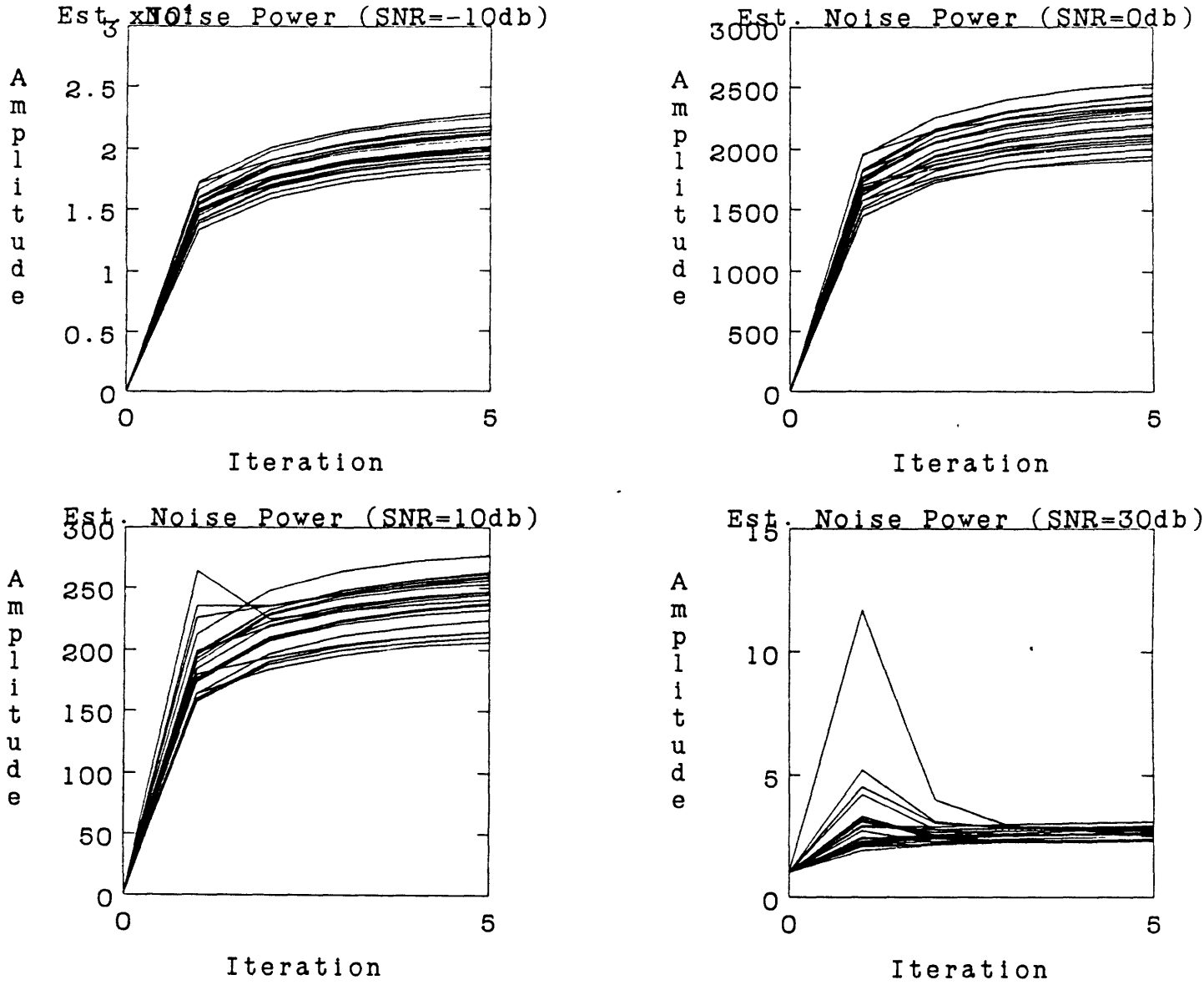


Figure 16e - EM-ML Delay/Gain Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

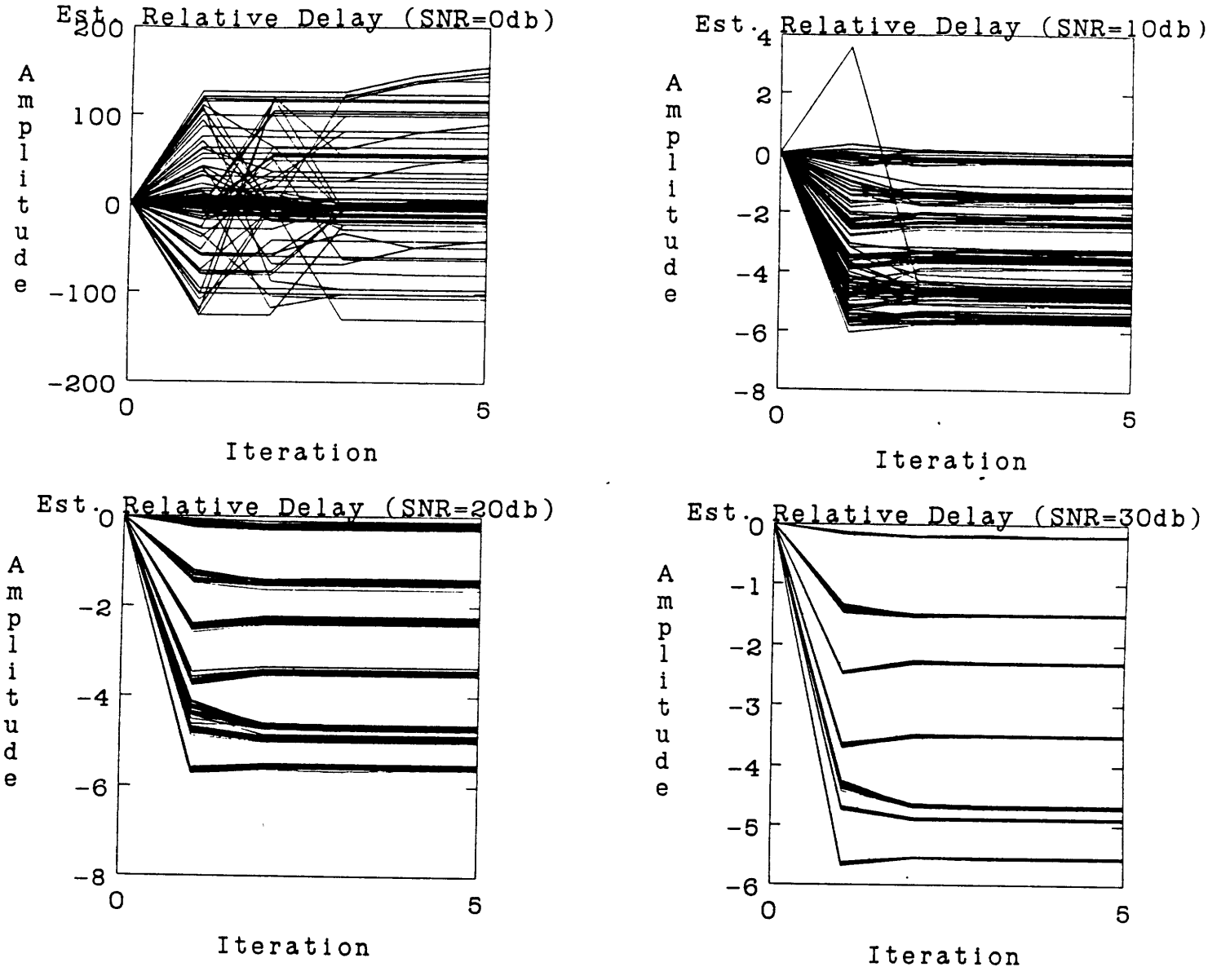


Figure 17a - EM-ML Delay/Gain Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 256$ point data, $M = 8$ channels, $\bar{\underline{\tau}} = (\,-4.7\ \ -4.9\ \ -1.5\ \ -3.5\ \ -5.55\ \ \ -.2\ \ -2.3\ \ 0\,)^T$, $\underline{\alpha} = (2\ \ 3\ \ 3\ \ 3\ \ 5\ \ 5\ \ 5\ \ 5)^T$, 20 runs for each SNR, estimate 8 delays, 8 gains, 10 pole signal power spectrum with gain, 8 noise level parameters.

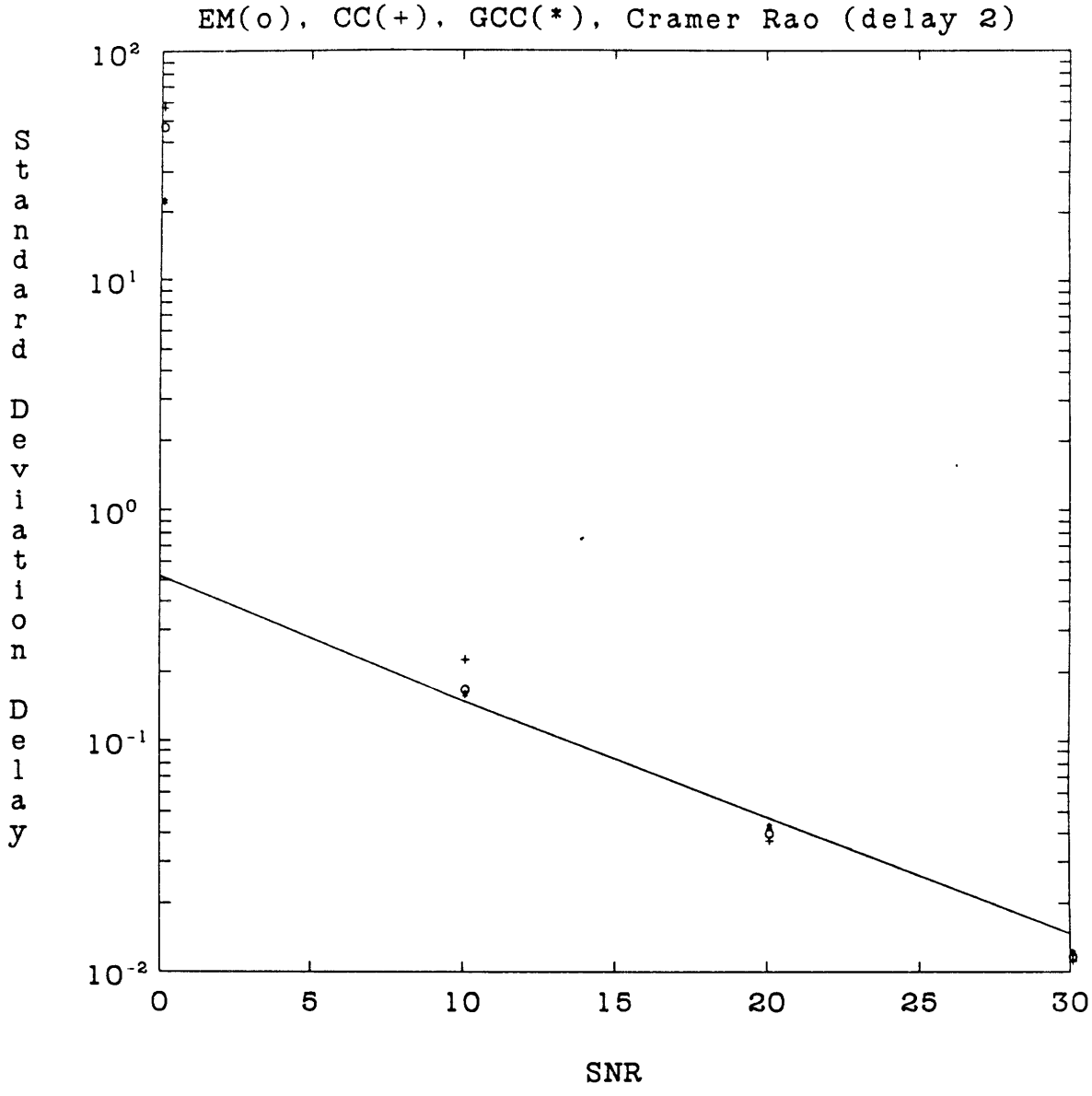


Figure 17b - EM-ML Delay/Gain Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_8^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

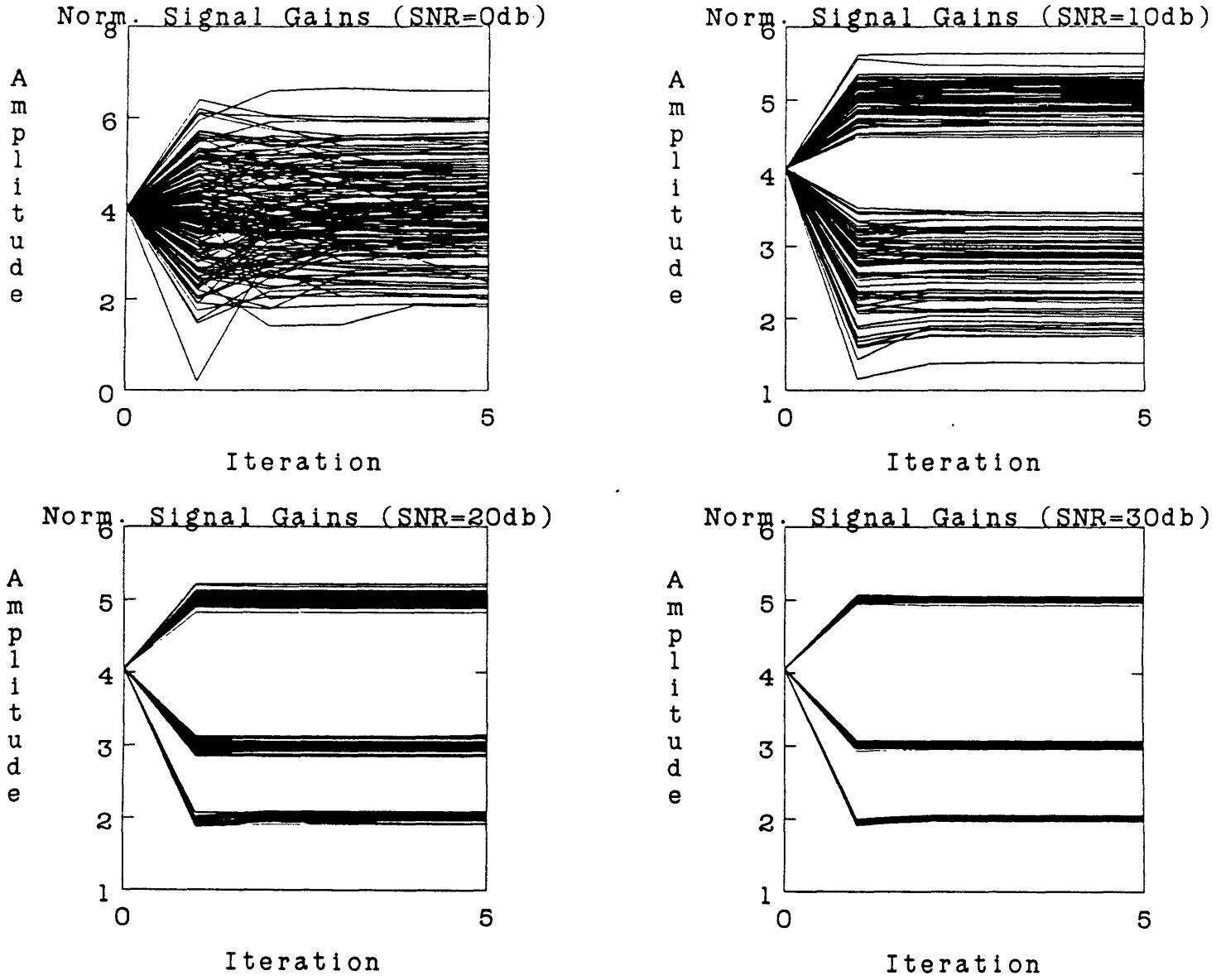


Figure 17c - EM-ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

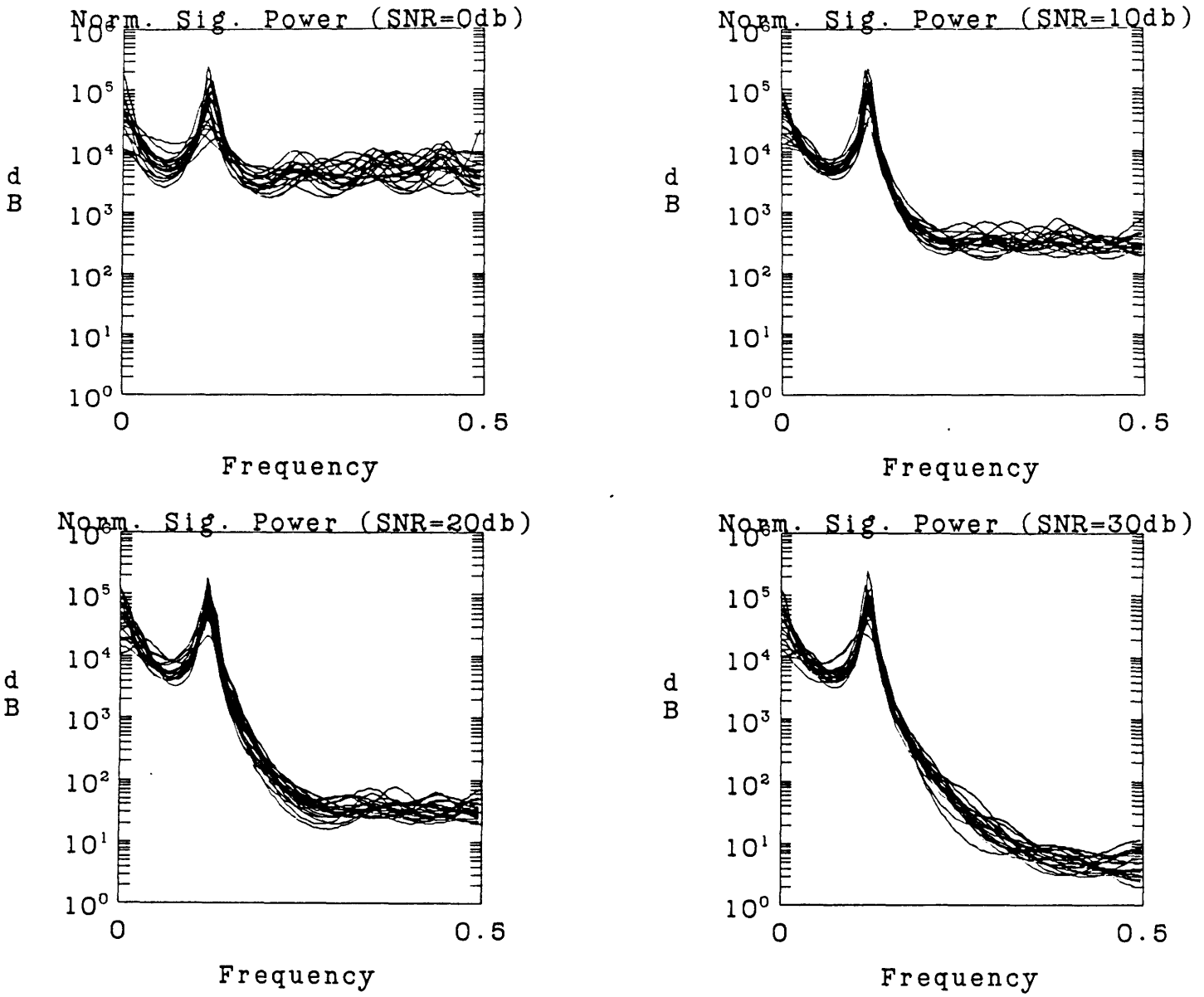


Figure 17d - EM-ML Delay/Gain Algorithm, Normalized Signal Power Spectra $P_S(\omega_n; \hat{\underline{\theta}}^{(l)}) \sum_{i=1}^{M} \hat{\alpha}_i^{(l)^2}$.

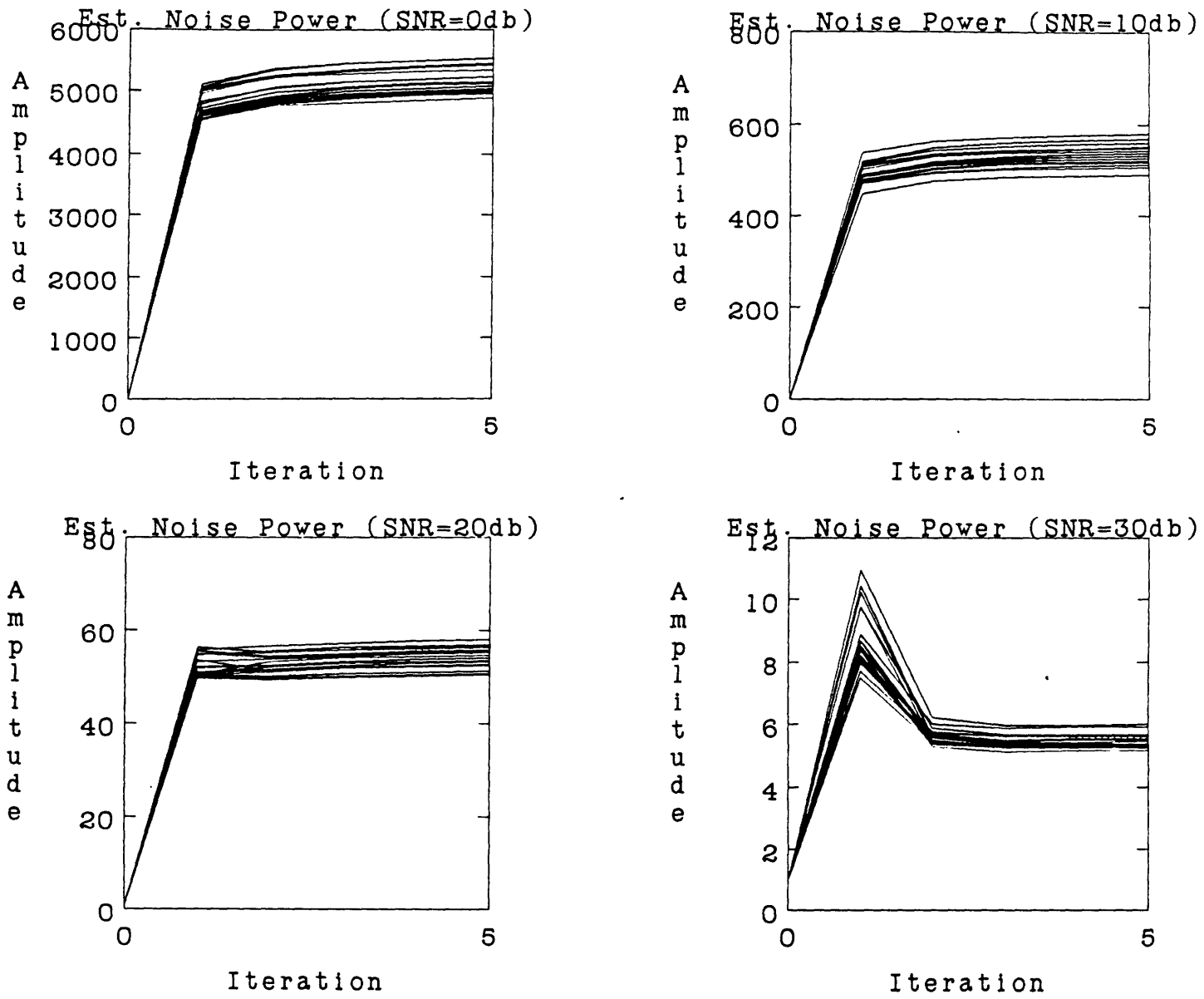


Figure 17e - EM-ML Delay/Gain Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

estimation algorithm is therefore not that clear, particularly given the high computational overhead of solving an eigenvector problem for the signal gains.

7.5 ML Coordinate Ascent Algorithm

Finally, we ran the same simulations on the ML coordinate ascent algorithm. Each of the M delays was estimated individually by maximizing the likelihood function. The M signal gains and a single noise level were estimated by directly maximizing the likelihood, which requires solving an eigenvector problem. An unparameterized signal power spectrum was also estimated by direct likelihood maximization. Samples of this spectral estimate which were negative or very small, were set to a small positive level determined as a small fraction of the noise level. Figures 18 show the performance on the $M = 2$ example, figures 19 show the $M = 3$ example, and figures 20 show the $M = 8$ example. In all cases, we estimated M delays, M gains, 1 noise level, and the signal spectrum. Behavior of the delay, signal gain, and noise level parameter estimates is virtually the same in all cases as for the EM-ML delay/gain algorithm, thus suggesting that getting smooth power spectra estimates is not essential for good estimation of the remaining parameters. (Note the ragged shape of the power spectra estimates, caused by using a periodogram-like technique to estimate this spectrum.) The advantage of this algorithm over the EM-ML delay/gain algorithm is that we avoid fitting an AR model to the signal periodogram. However, the computation involved in fitting an AR model is negligible compared with the time required to estimate the delays. Therefore, in practice the advantage of direct ML maximization over all parameters is not clear.

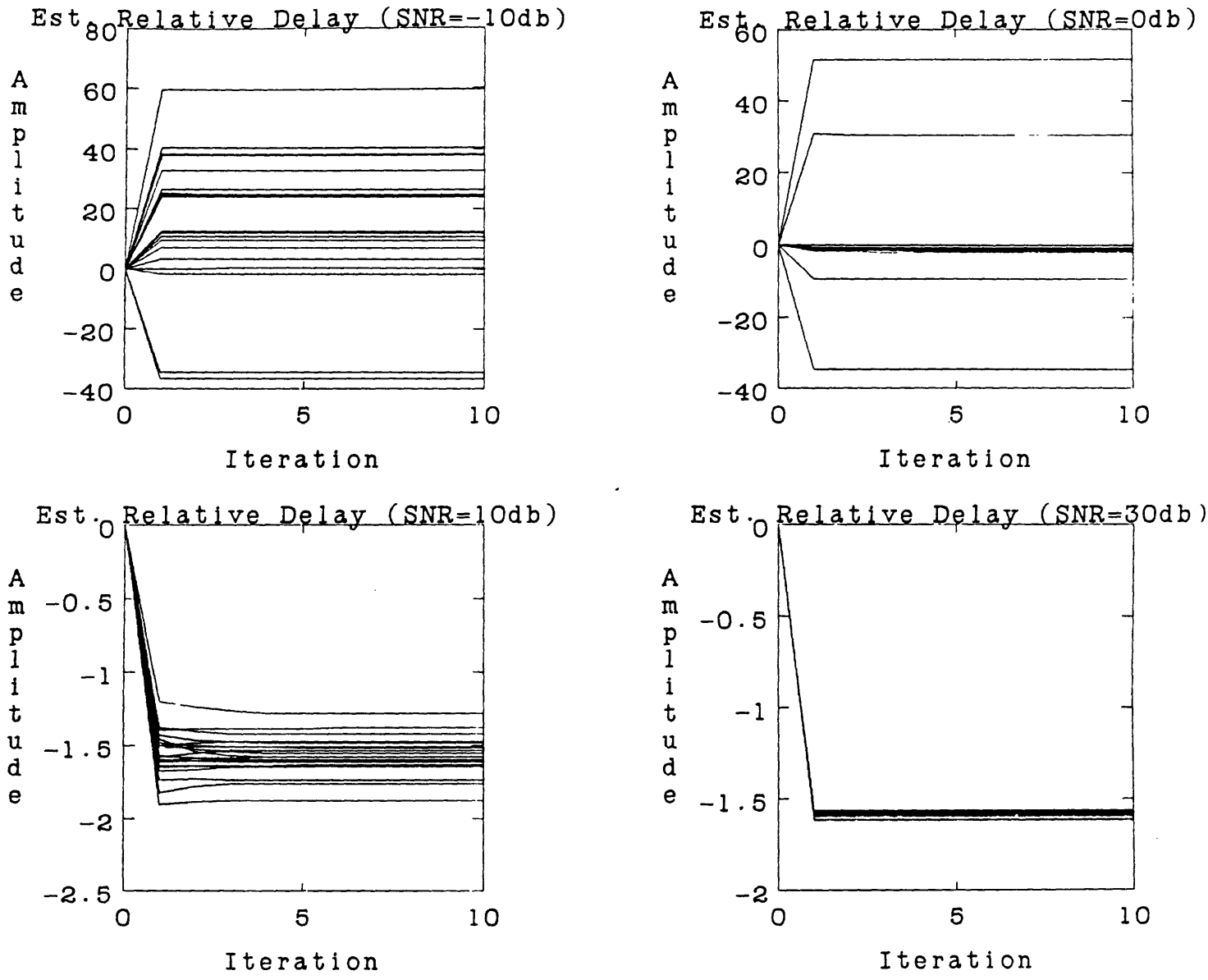


Figure 18a - ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 2$ channels, $\bar{\tau}_1 - \bar{\tau}_2 = -1.58$, $\underline{\bar{\alpha}} = (3\ \ 3)^T$, 20 runs for each SNR, estimate 2 delays, 2 gains, 6 pole signal power spectrum with gain, 2 noise level parameters.

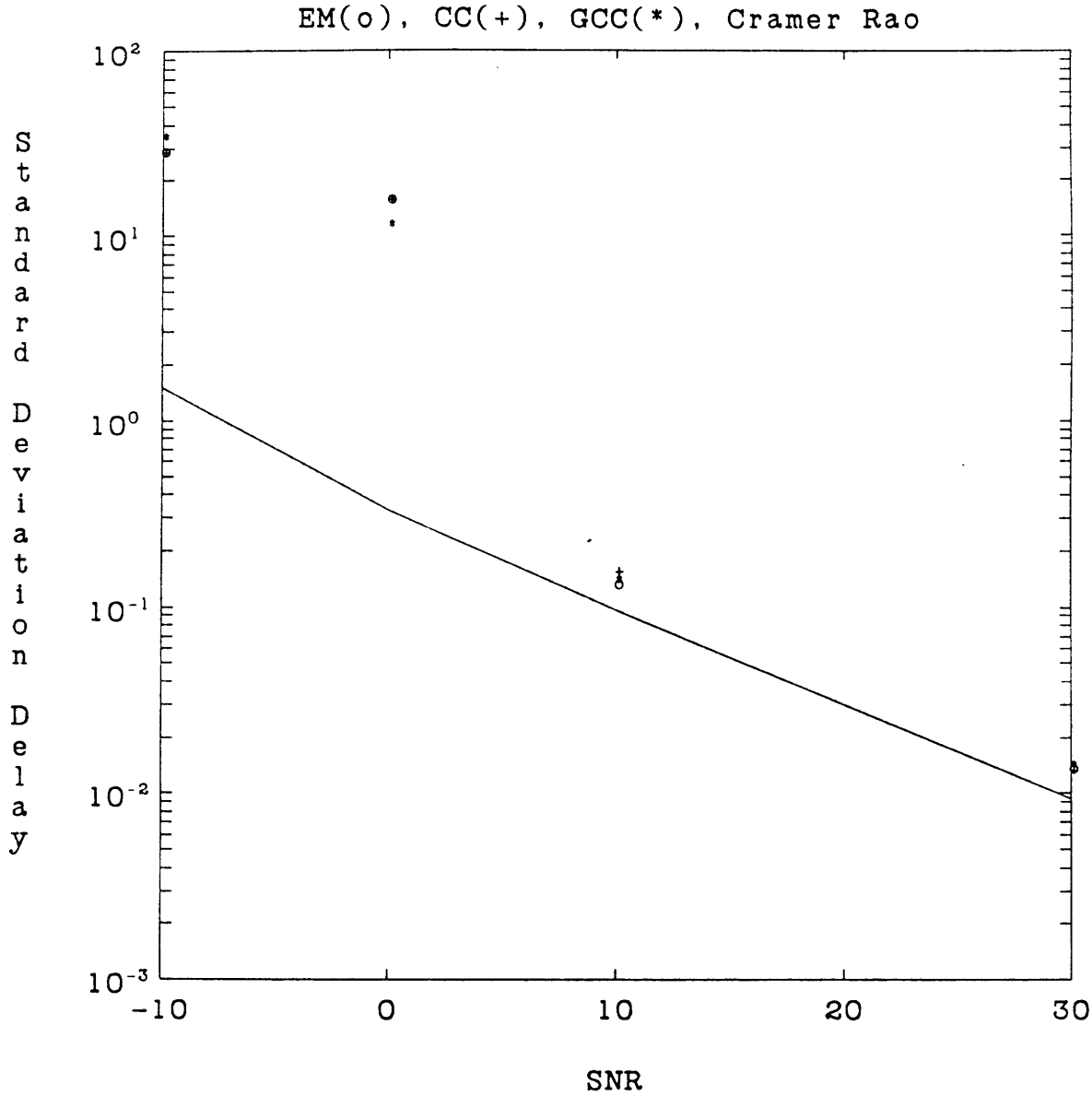


Figure 18b - ML Algorithm, Standard Deviation of Relative Delay Estimates - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

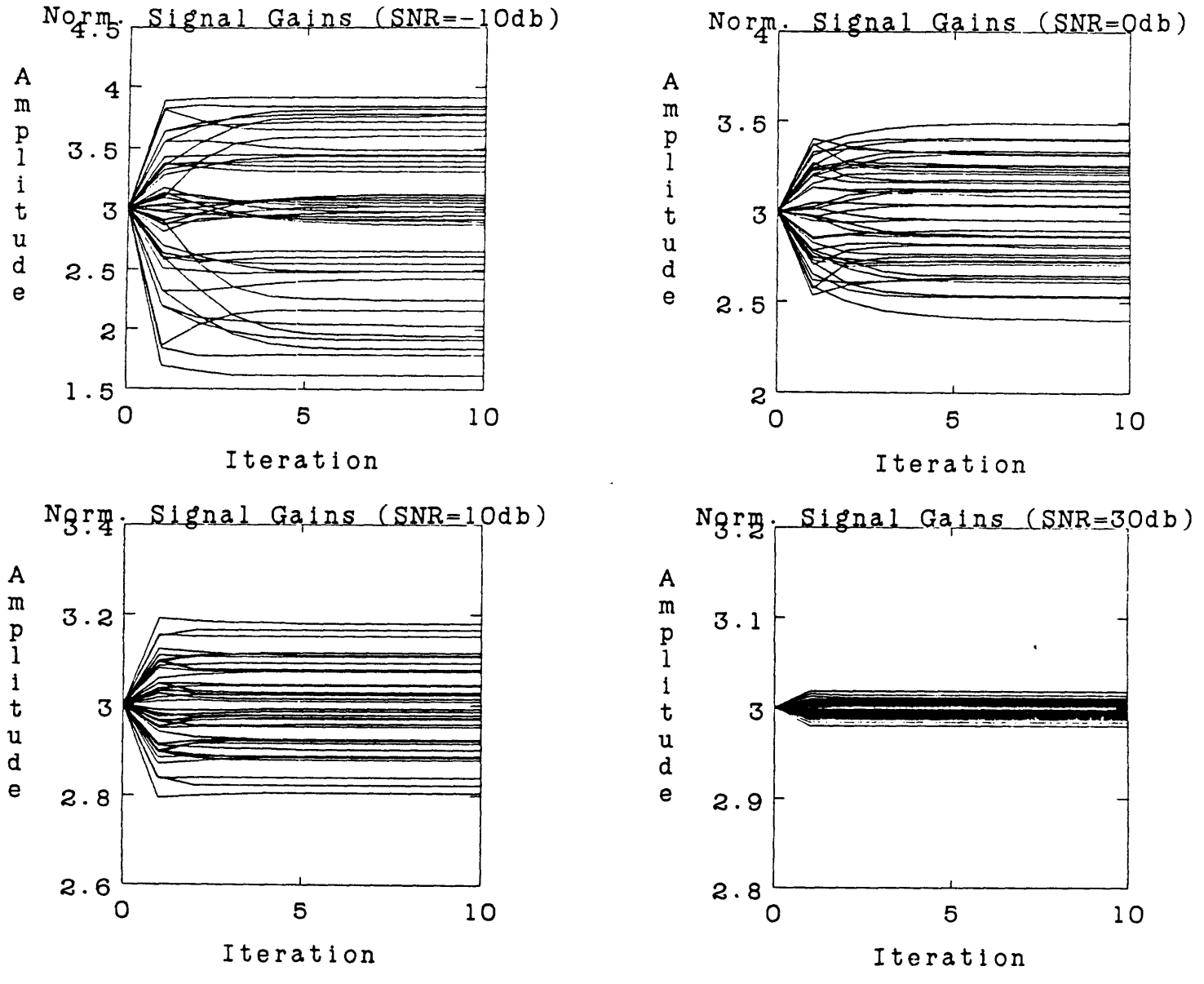


Figure 18c - ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

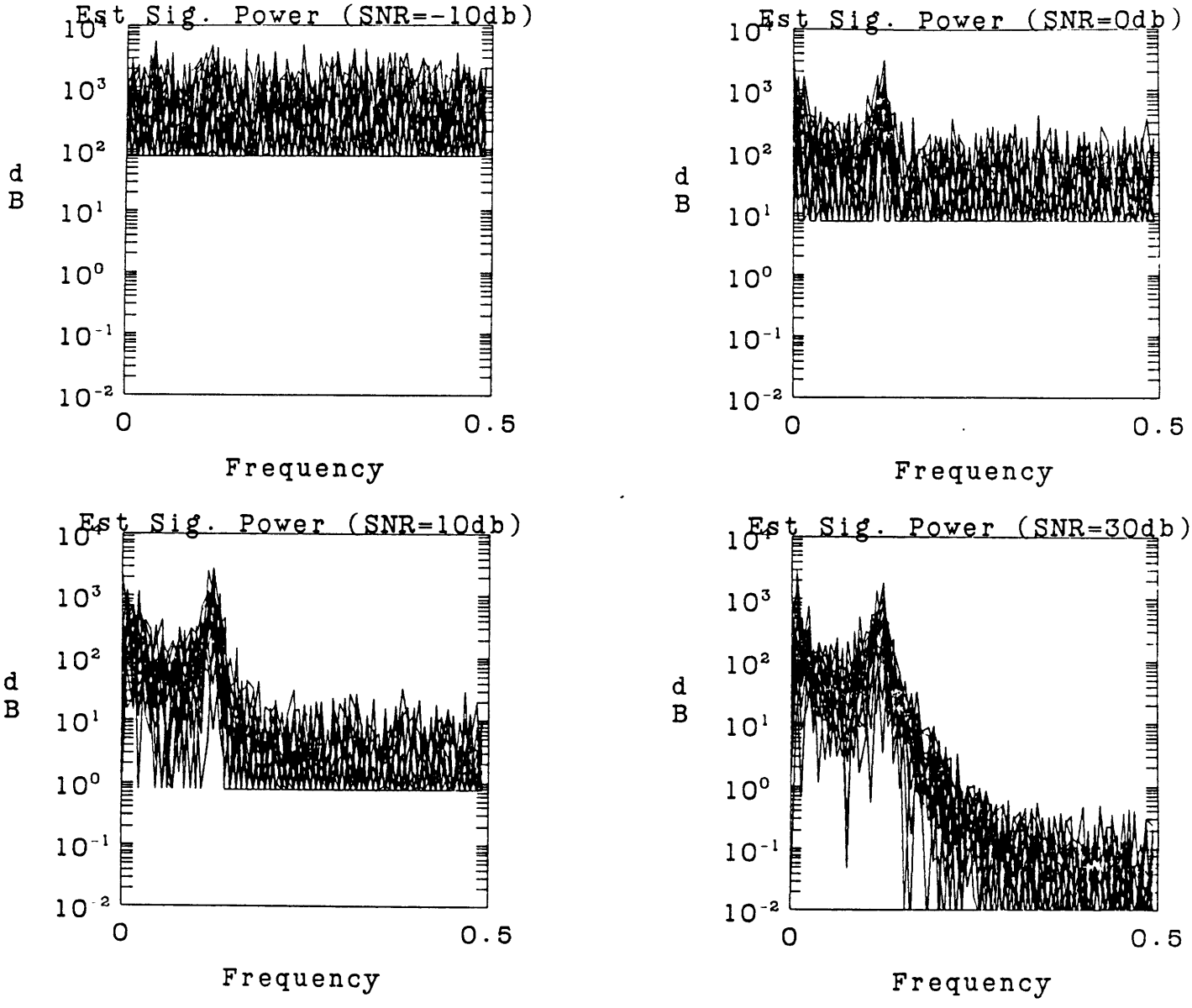


Figure 18d - ML Algorithm, Signal Power Spectra $\hat{P}_S^{(l)}(\omega_n)$.

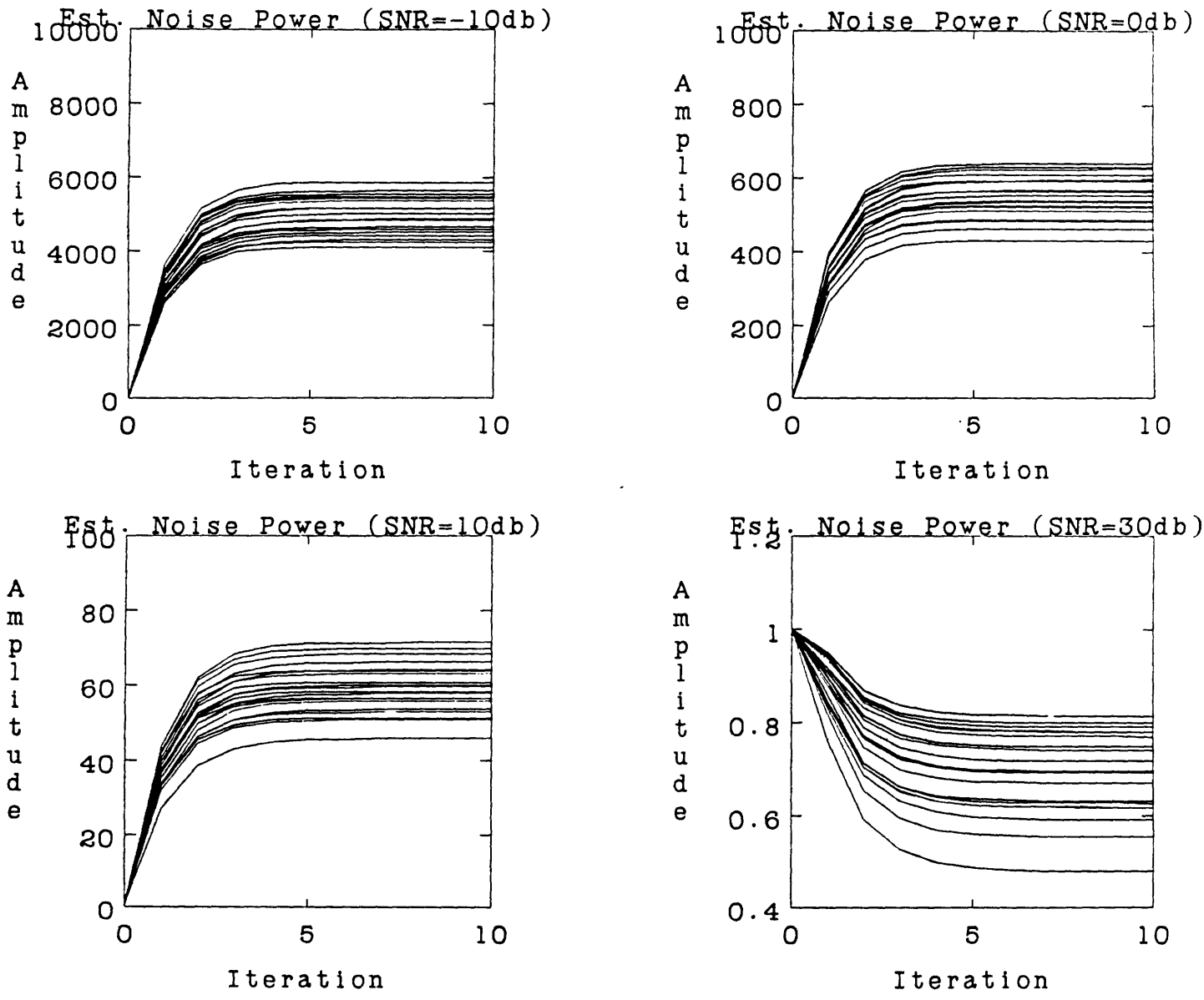


Figure 18e - ML Algorithm, Noise Level Estimates $\hat{\sigma}^{(\iota)}$

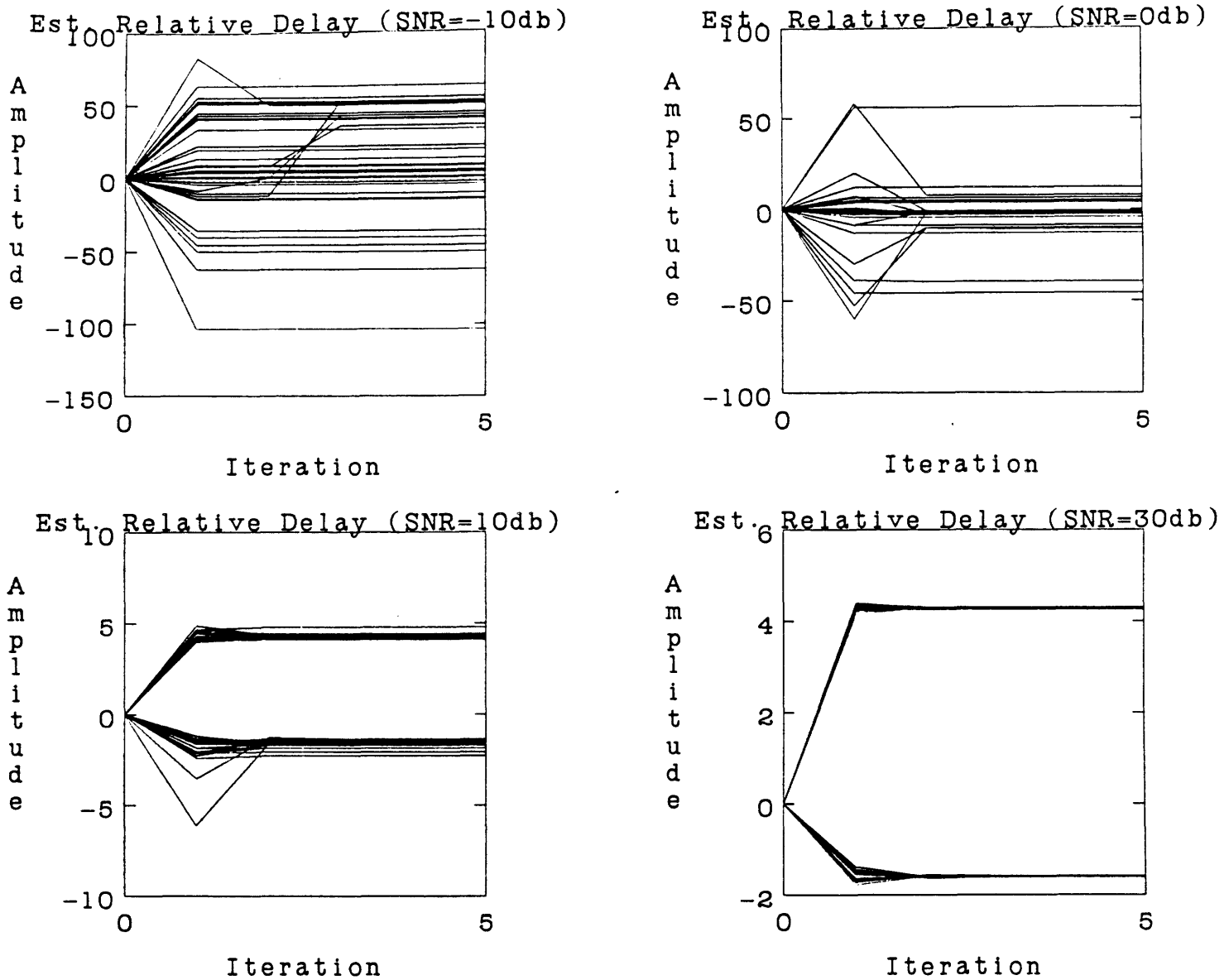


Figure 19a - ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 128$ point data, $M = 3$ channels, $\underline{\bar{\tau}} = (-1.58 \;\; 4.29 \;\; 0)^T$, $\underline{\bar{\alpha}} = (3 \;\; 4.5 \;\; 6)^T$, 20 runs for each SNR, estimate 3 delays, 3 gains, 6 pole signal power spectrum with gain, 3 noise level parameters.

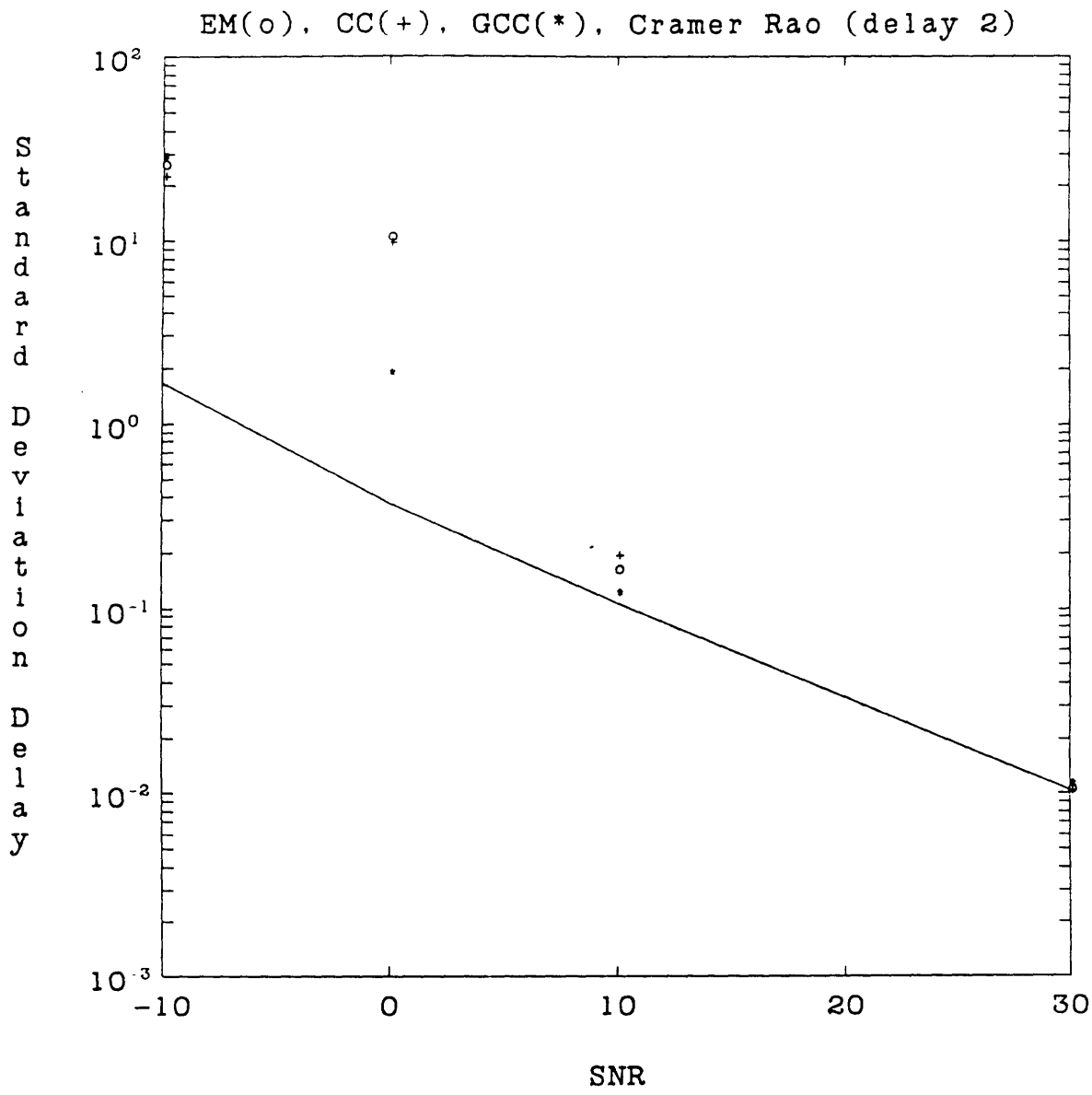


Figure 19b - ML Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_3^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

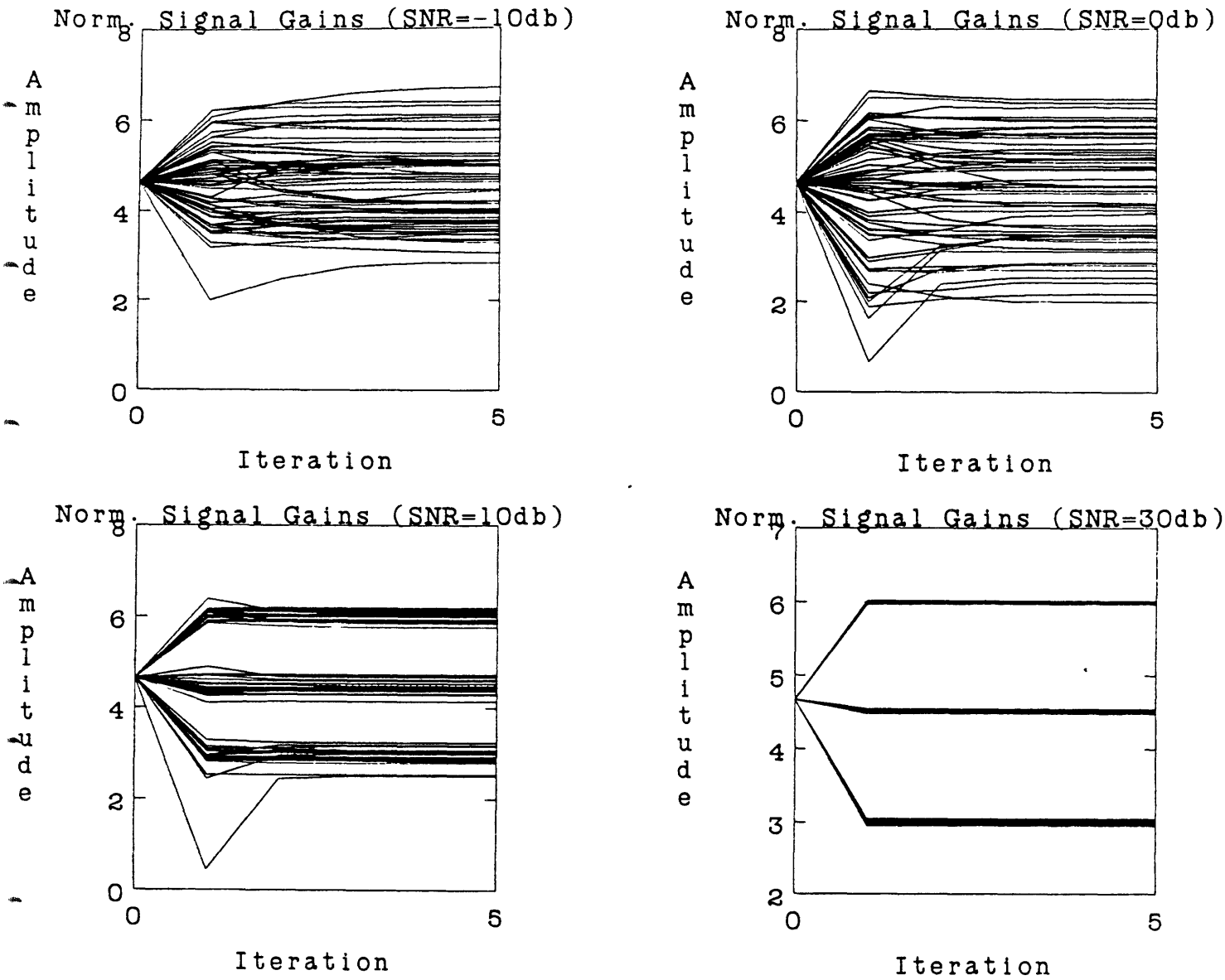


Figure 19c - ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

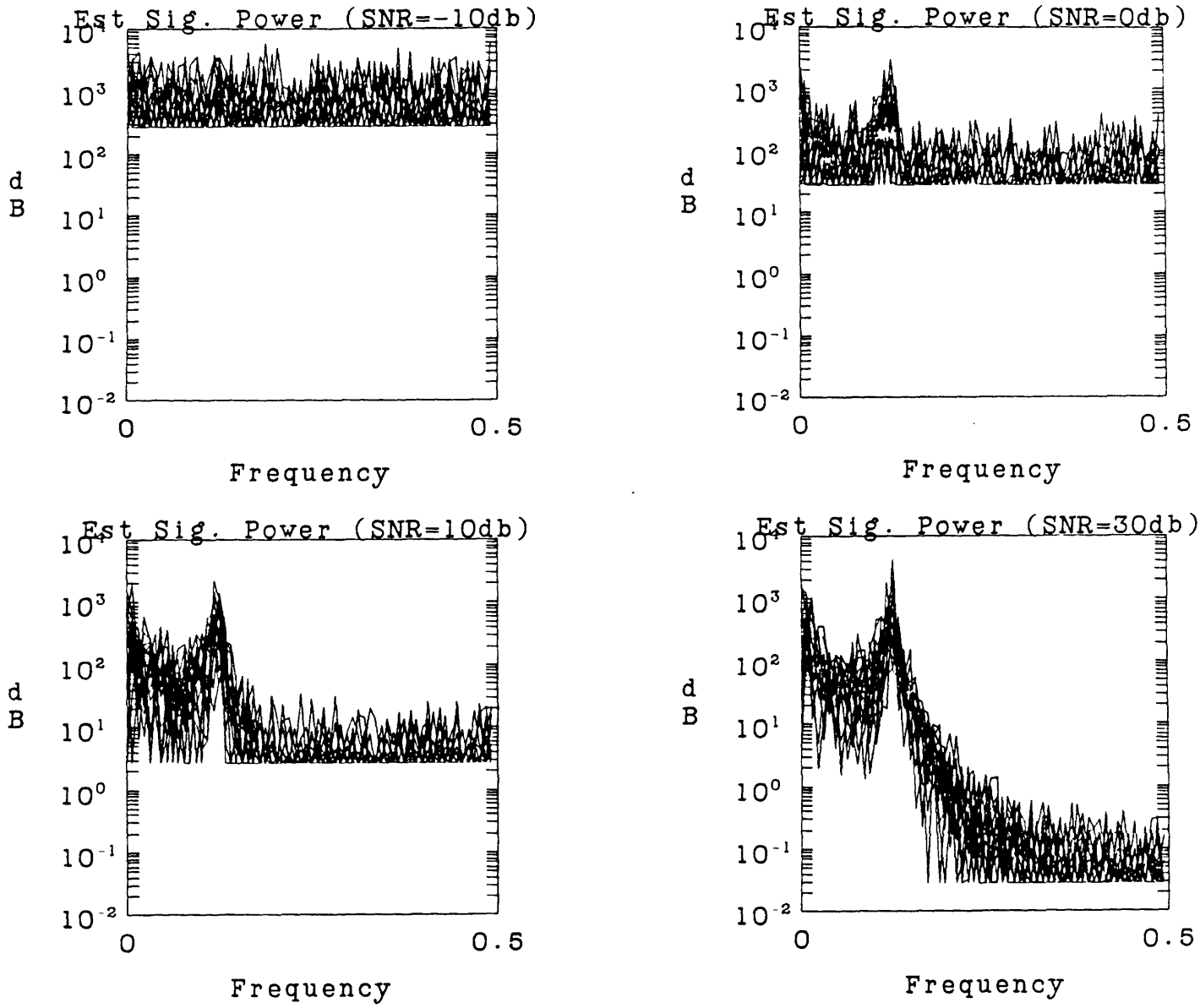


Figure 19d - ML Algorithm, Signal Power Spectra $\hat{P}_S^{(l)}(\omega_n)$.

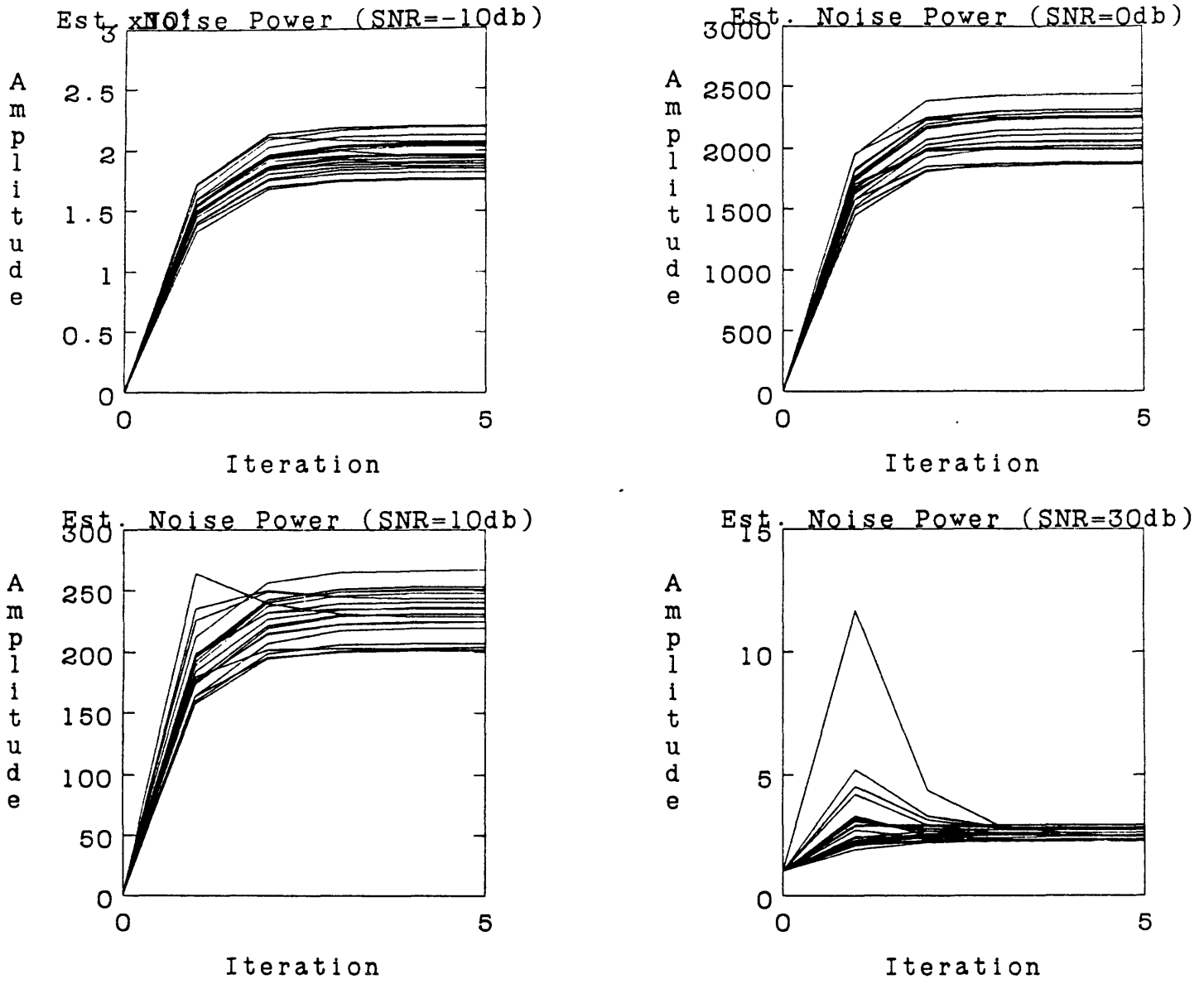


Figure 19e - ML Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

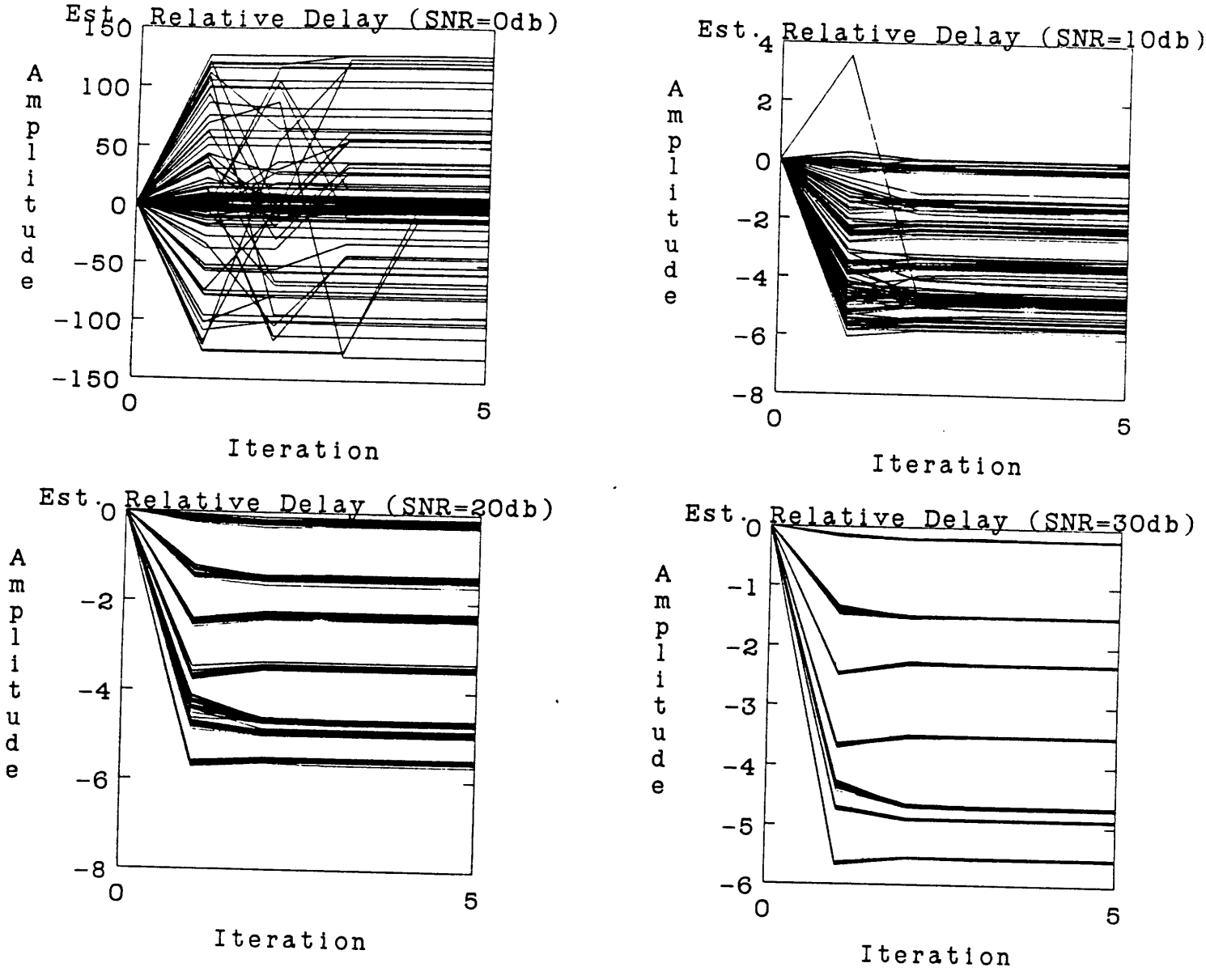


Figure 20a - ML Algorithm, Relative Delay Estimates $\hat{\underline{\tau}}^{(l)}$ - unknown lowpass signal model, unknown noise levels, $N = 256$ point data, $M = 8$ channels, $\underline{\tau} = (\text{-4.7 -4.9 -1.5 -3.5 -5.55 -.2 -2.3 0})^T$, $\underline{\alpha} = (2\ 3\ 3\ 3\ 5\ 5\ 5\ 5)^T$, 20 runs for each SNR, estimate 8 delays, 8 gains, 10 pole signal power spectrum with gain, 8 noise level parameters.

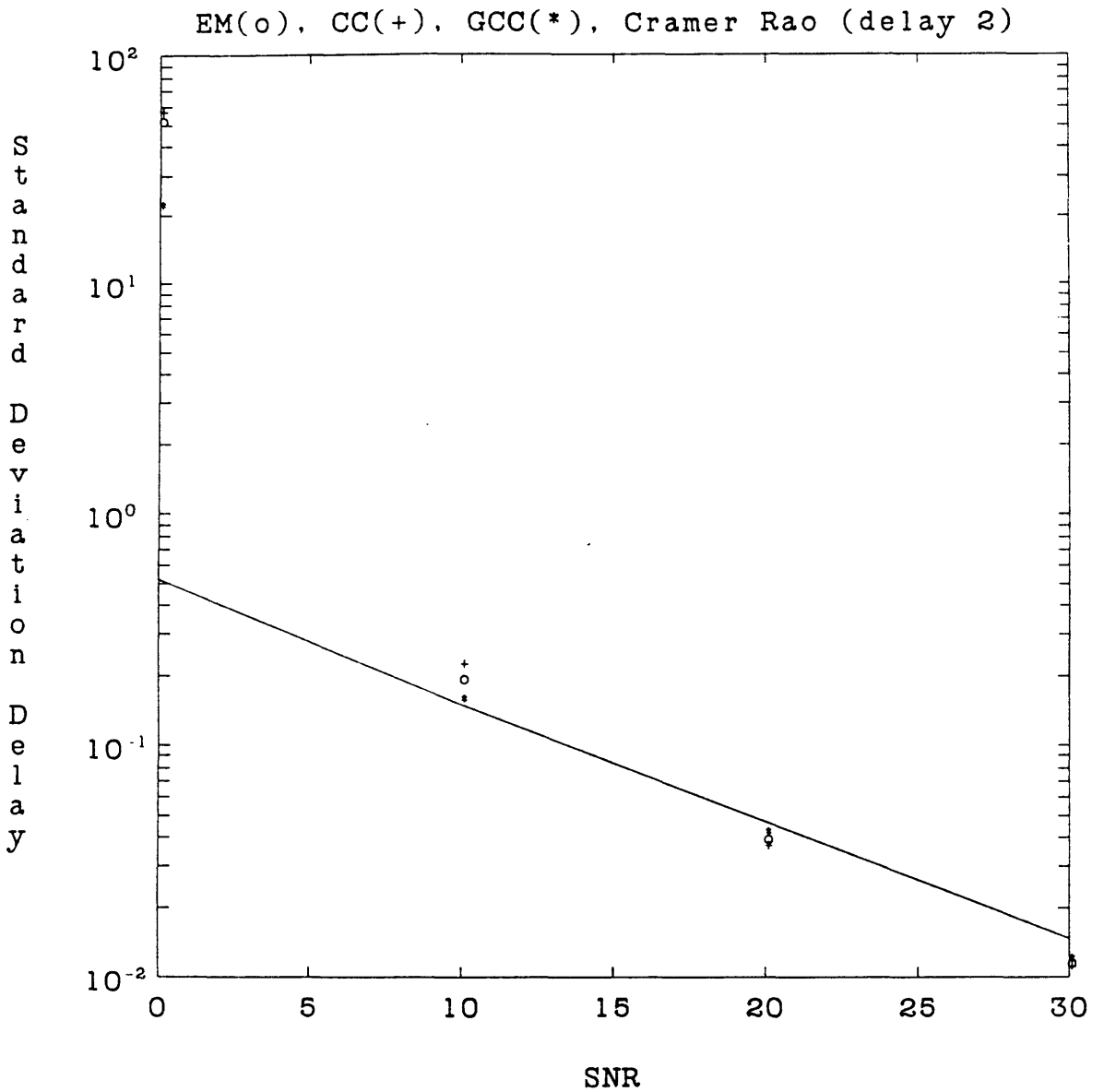


Figure 20b - ML Algorithm, Standard Deviation of Relative Delay Estimates $\hat{\underline{\tau}}_2^{(l)} - \hat{\underline{\tau}}_8^{(l)}$ - EM(o), Cross Correlation Method (CC)(+), Generalized Cross Correlation Method (GCC)(*) vs. Cramer-Rao lower bound.

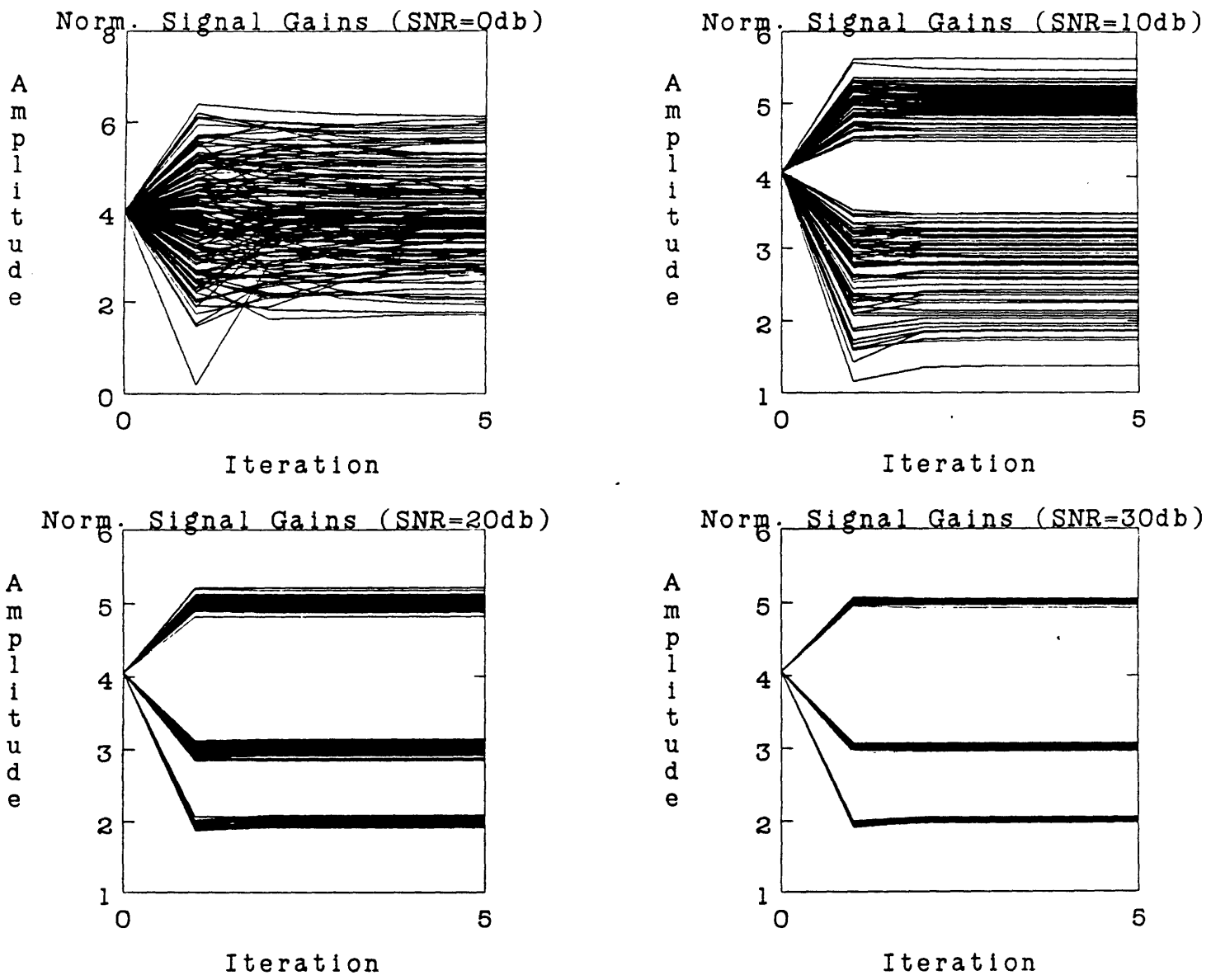


Figure 20c - ML Algorithm, Normalized Gain Estimates $\hat{\underline{\alpha}}^{(l)}/\kappa$.

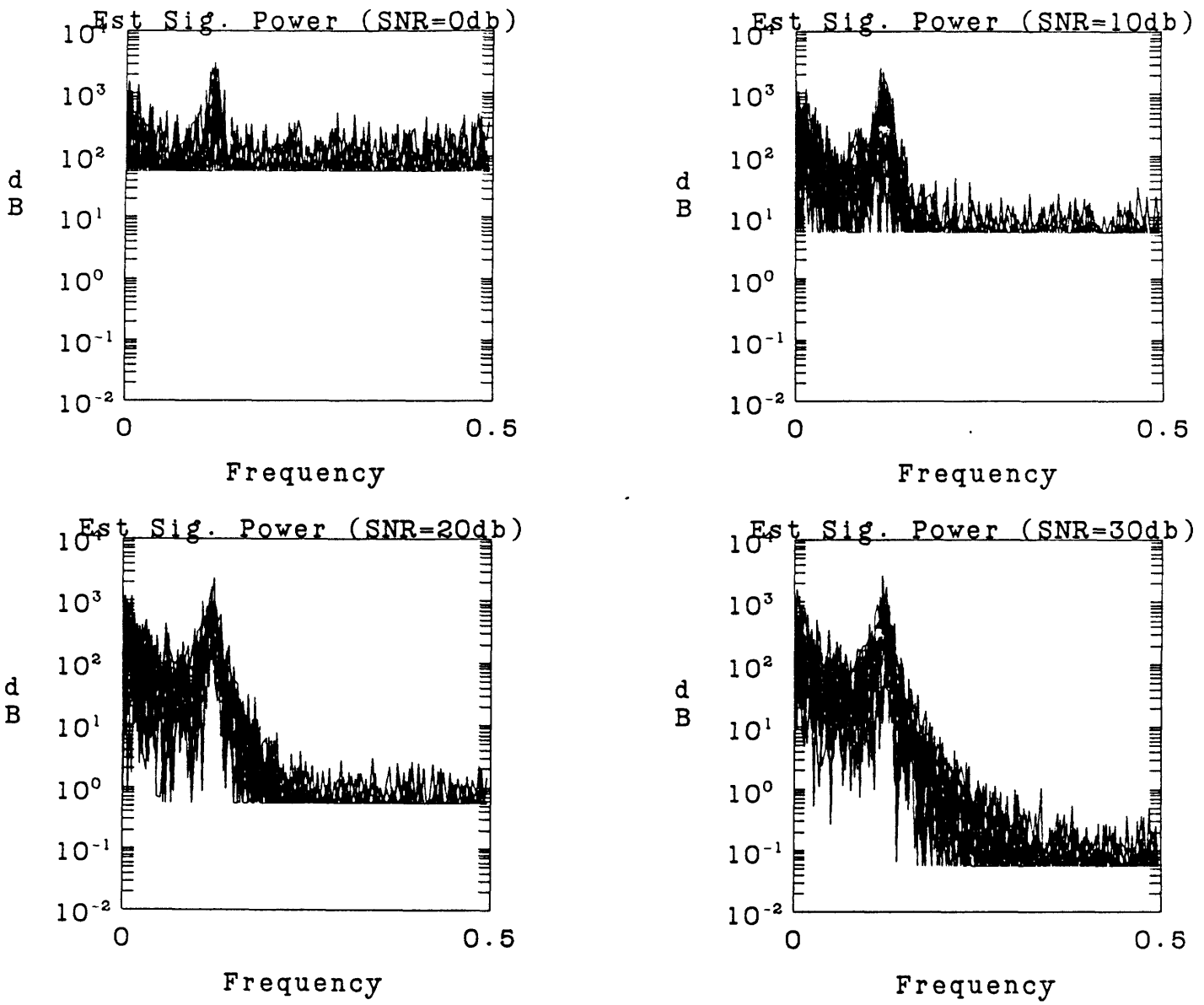


Figure 20d - ML Algorithm, ML Algorithm, Signal Power Spectra $\hat{P}_S^{(l)}(\omega_n)$.

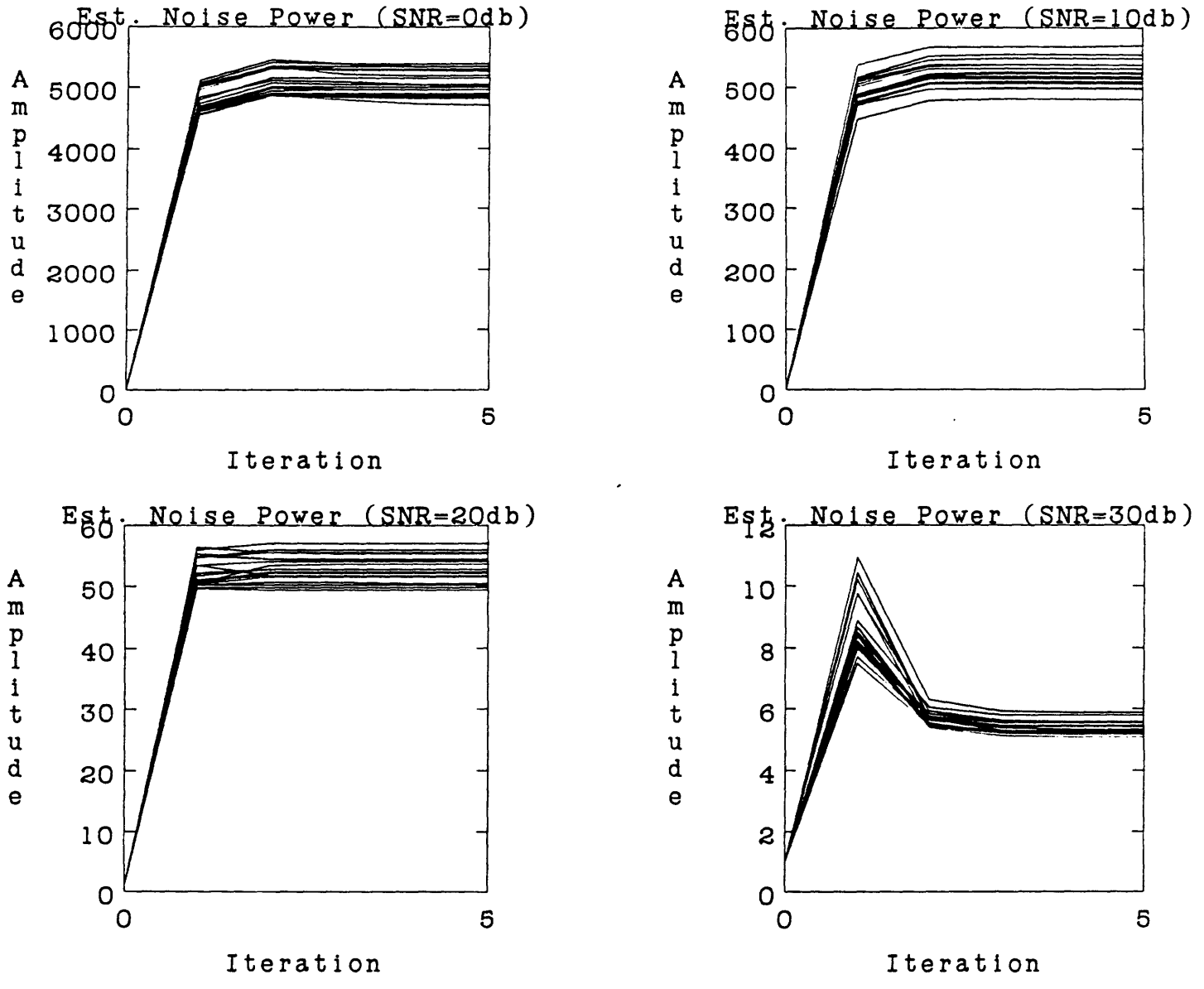


Figure 20e - ML Algorithm, Noise Level Estimates $\hat{\sigma}^{(l)}$

8 Delay and Doppler Estimation

If the source is moving relative to the array, the signals observed at different receivers are not only time delayed, but also time compressed relative to each other. Measurement of these differential Doppler time compression coefficients can provide important additional information concerning source location, velocity, and heading.

The signal model is now given by:

$$z_i(t) = \alpha_i s\left(t - \tau_i(t)\right) + v_i(t) \qquad \text{for } i = 1, 2, \ldots, M \tag{138}$$

where $\tau_i(t)$ are the time-varying delays caused by the relative motion between the source and the receivers. To simplify the exposition, we shall make the following assumptions.

(1) Assume that the change in the array-source geometry during the observation interval is small, so that the various $\tau_i(t)$ are essentially linearly time varying:

$$\tau_i(t) = \tau_i + \dot{\tau}_i t \tag{139}$$

where τ_i is the delay at $t = 0$, and $\dot{\tau}_i = d\tau_i(t)/dt$ is the delay derivative, or Doppler coefficient. Substituting (139) into (138), we observe that the Doppler effect results in a time-scaling or compression of the received signals. We note that $\dot{\tau}_i = \nu_i/c$ where ν_i is the radial component of source velocity towards the i^{th} receiver, and c is the velocity of propagation in the medium. Thus, estimation of the $\dot{\tau}_i$ is equivalent to the estimation of the ν_i. In most situations of practical interest $\nu_i/c \ll 1$, so that the expected Doppler $\dot{\tau}_i \ll 1$.

(2) Taking the Fourier transform of $z_i(t)$ gives:

$$Z_i(\omega_n) = \alpha_i e^{-j\omega_n \tau_i} S\left(\frac{\omega_n}{1 - \dot{\tau}_i}\right) + V_i(\omega_n) \tag{140}$$

This equation suggests that the Doppler shift causes signal components at higher frequencies to shift more in frequency than components at lower frequencies. To avoid having to compensate for this effect, let us assume that $s(t)$ is narrowband, centered at frequency ω_0 with bandwidth W_S about the center frequency, where $W_S/\omega_0 \ll 1$. Let us also assume that $W_S\dot{\tau}_i < 2\pi/T$, so that difference between the minimal and maximal Doppler shift over the signal modulation bandwidth does not exceed a full cycle (2π

radians) over the observation period T. Under these conditions, the Doppler shift is approximately equivalent to a frequency shift of the signal waveform. Thus:

$$Z_i(\omega_n) \approx \alpha_i e^{-j\omega_n \tau_i} S\left(\omega_n + \Delta\omega_i\right) + V_i(\omega_n) \tag{141}$$

where $\Delta\omega_i$ are the Doppler frequency shifts:

$$\Delta\omega_i = \omega_0 \dot{\tau}_i \tag{142}$$

(3) In order to simplify the use of finite length FFT's, we shall confine the $\Delta\omega_i$ to the set:

$$\Delta\omega_i = \frac{2\pi}{T} i \qquad i = 0, \pm 1, \pm 2, \ldots \tag{143}$$

For $\omega_0 T/2\pi \gg 1$, the interval $2\pi/T$ is a very small fraction of the carrier frequency, and this assumption is not very restrictive. (If necessary, the data could be lengthened by zero padding to allow use of longer FFT's and denser sampling in the frequency domain.)

(4) We will assume that the noises $v_i(t)$ are wideband relative to the signal $s(t)$ and have relatively smooth power spectra relative to the signal $s(t)$ over a frequency range equal to the maximum expected Doppler shifts, so that:

$$P_{V_i}(\omega + \Delta\omega_i; \underline{\sigma}_i) \approx P_{V_i}(\omega; \underline{\sigma}_i) \tag{144}$$

We note that these assumptions are frequently satisfied in passive sonar applications.

Rewriting (141):

$$Z_i(\omega_n - \Delta\omega_i) = \alpha_i e^{-j(\omega_n - \Delta\omega_i)\tau_i} S(\omega_n) + V_i(\omega_n - \Delta\omega_i) \tag{145}$$

Let us define:

$$\underline{Z}(\omega_n) = \begin{pmatrix} Z_1(\omega_n - \Delta\omega_1) \\ Z_2(\omega_n - \Delta\omega_2 \\ \vdots \\ Z_M(\omega_n - \Delta\omega_M) \end{pmatrix} \tag{146}$$

Since we have assumed that $\Delta\omega_i$ is a multiple integer of the fundamental frequency $\Delta\omega_i = 2\pi/T$, then (145) implies that different frequency samples of $\underline{Z}(\omega_n)$ are statistically uncorrelated with each other. Let $\underline{\xi}$ be the vector of parameters to be estimated:

$$\underline{\xi} = \begin{pmatrix} \underline{\tau} \\ \underline{\Delta\omega} \\ \underline{\alpha} \\ \underline{\theta} \\ \underline{\sigma} \end{pmatrix} \tag{147}$$

where $\underline{\Delta\omega}$ is the $M \times 1$ vector of $\Delta\omega_i$'s. Then the covariance matrix associated with $\underline{Z}(\omega_n)$ is:

$$\begin{aligned} \mathbf{P}_Z(\omega_n;\underline{\xi}) &= \mathrm{E}_{\underline{\xi}}\left[\underline{Z}(\omega_n)\underline{Z}^*(\omega_n)\right] \\ &= P_S(\omega_n;\underline{\theta})\underline{U}(\omega_n;\underline{\xi})\underline{U}^*(\omega_n;\underline{\xi}) + \mathbf{P}_V(\omega_n;\underline{\sigma}) \end{aligned} \tag{148}$$

where $\mathbf{P}_V(\omega_n;\underline{\sigma})$ is defined in (11), and:

$$\underline{U}(\omega_n;\underline{\xi}) = \begin{pmatrix} \alpha_1 e^{-j(\omega_n-\Delta\omega_1)\tau_1} \\ \vdots \\ \alpha_M e^{-j(\omega_n-\Delta\omega_M)\tau_M} \end{pmatrix} \tag{149}$$

The observed data log-likelihood is given by:

$$L_Z(\underline{\xi}) = -\sum_{n=L}^{U}\left[\log\det\mathbf{P}_Z(\omega_n;\underline{\xi}) + \underline{Z}^*(\omega_n)\mathbf{P}_Z^{-1}(\omega_n;\underline{\xi})\underline{Z}(\omega_n)\right] \tag{150}$$

Substituting (146) and (148) into (150) and carrying out the indicated matrix manipulations, we obtain an expression similar to (13):

$$\begin{aligned} L_Z(\underline{\xi}) = c - \sum_{n=L}^{U}\Bigg[\sum_{i=1}^{M}\log P_{V_i}(\omega_n;\underline{\sigma}_i) + \log\left(1+\sum_{i=1}^{M}\frac{\alpha_i^2 P_S(\omega_n;\underline{\theta})}{P_{V_i}(\omega_n;\underline{\sigma}_i)}\right) \\ + \sum_{i=1}^{M}\frac{|Z_i(\omega_n-\Delta\omega_i)|^2}{P_{V_i}(\omega_n;\underline{\sigma}_i)} - \frac{P_S(\omega_n;\underline{\theta})\left|\sum_{i=1}^{M}\alpha_i e^{-j(\omega_n-\Delta\omega_i)\tau_i}Z_i^*(\omega_n-\Delta\omega_i)/P_{V_i}(\omega_n;\underline{\sigma}_i)\right|^2}{1+\sum_{i=1}^{M}\alpha_i^2 P_S(\omega_n;\underline{\theta})/P_{V_i}(\omega_n;\underline{\sigma}_i)}\Bigg] \end{aligned} \tag{151}$$

where c is independent of $\underline{\xi}$.

The presence of the Doppler shifts complicates the estimation problem quite drastically. Even if the signal gains and spectral parameters are precisely known, the direct

ML approach requires maximizing with respect to $\underline{\tau}$ and $\underline{\Delta\omega}$ jointly, leading to a complex multi-dimensional search.

Careful examination of (151) indicates that, under our assumption that all spectra are smooth, we can only estimate relative delays $\tau_i - \tau_j$ and relative Dopplers $\Delta\omega_i - \Delta\omega_j$ from the observations (see [19,18]). Thus, as in the stationary case, we could fix one delay and one Doppler to an arbitrary value, say $\tau_M = 0$ and $\Delta\omega_M = 0$, and estimate only the remaining $(M-1)$ delays and $(M-1)$ Dopplers.

8.1 EM Algorithm for Relative Doppler Estimation

Consider first the application of the EM algorithm. Here we iteratively compute:

For $l = 0, 1, \ldots$

$$\hat{\underline{\tau}}^{(l+1)}, \widehat{\underline{\Delta\omega}}^{(l+1)}, \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\theta}}^{(l+1)} \leftarrow \max_{\underline{\tau}, \underline{\Delta\omega}, \underline{\alpha}, \underline{\theta}} \mathrm{E}_{\underline{\xi}^{(l)}} \left[\log p \left(\underline{s}, \underline{z}; \underline{\tau}, \underline{\Delta\omega}, \underline{\alpha}, \underline{\theta}, \hat{\underline{\sigma}}^{(l)} \right) \middle| \underline{z} \right] \quad (152)$$

$$\hat{\underline{\sigma}}^{(l+1)} \leftarrow \max_{\underline{\sigma}} \mathrm{E}_{\underline{\xi}^{(l)}} \left[\log p \left(\underline{s}, \underline{z}; \hat{\underline{\tau}}^{(l+1)}, \widehat{\underline{\Delta\omega}}^{(l+1)}, \hat{\underline{\alpha}}^{(l+1)}, \hat{\underline{\theta}}^{(l+1)}, \underline{\sigma} \right) \middle| \underline{z} \right] \quad (153)$$

Following the development of the previous section, we obtain in complete analogy with (58), (59), (62), (63), (64), and (65):

E – step :

$$\hat{S}^{(l)}(\omega_n) = \sum_{k=1}^{M} \hat{\alpha}_k^{(l)} e^{j(\omega_n - \widehat{\Delta\omega}_i^{(l)})\hat{\tau}_k^{(l)}} Z_k(\omega_n - \widehat{\Delta\omega}_i^{(l)}) \hat{T}^{(l)}(\omega_n) / P_{V_k}(\omega_n; \underline{\sigma}_k) \quad (154)$$

$$\widehat{|S(\omega_n)|^2}^{(l)} = \left| \hat{S}^{(l)}(\omega_n) \right|^2 + \hat{T}^{(l)}(\omega_n) \quad (155)$$

$$\text{where } \hat{T}^{(l)}(\omega_n) = \frac{P_S(\omega_n; \hat{\underline{\theta}}^{(l)})}{1 + \sum_{k=1}^{M} \widehat{SNR}_k^{(l)}(\omega_n)}$$

M – step :

$$\hat{\underline{\theta}}^{(l+1)} \leftarrow \min_{\underline{\theta}} \sum_{n=L}^{U} \left[\log P_S(\omega_n; \underline{\theta}) + \frac{\widehat{|S(\omega_n)|^2}^{(l)}}{P_S(\omega_n; \underline{\theta})} \right] \quad (156)$$

For $i = 1, \ldots, M$

$$\hat{\tau}_i^{(l+1)}, \widehat{\Delta\omega}_i^{(l+1)} \leftarrow \max_{\tau_i, \Delta\omega_i} \left| \mathrm{Re} \left[\frac{\sum_{n=L}^{U} e^{-j(\omega_n - \Delta\omega_i)\tau_i} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n - \Delta\omega_i)}{P_{V_i}(\omega_n; \hat{\underline{\sigma}}_i^{(l)})} \right] \right| \quad (157)$$

$$\hat{\alpha}_i^{(l+1)} = \frac{\mathrm{Re}\left[\sum_{n=L}^{U} e^{-j(\omega_n - \widehat{\Delta\omega}_i^{(l+1)})\hat{\tau}_i^{(l+1)}} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n - \widehat{\Delta\omega}_i^{(l+1)}) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})\right]}{\sum_{n=L}^{U} |\widehat{S(\omega_n)}|^{2^{(l)}} / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})}$$

$$\underline{\hat{\sigma}}_i^{(l+1)} \leftarrow \min_{\underline{\sigma}} \left\{ \frac{\left| Z_i(\omega_n) - \hat{\alpha}_i^{(l+1)} e^{-j(\omega_n - \widehat{\Delta\omega}_i^{(l+1)})\hat{\tau}_i^{(l+1)}} \hat{S}^{(l)}(\omega_n) \right|^2 + \hat{\alpha}_i^{(l+1)^2} \hat{T}^{(l)}(\omega_n)}{P_{V_i}(\omega_n; \underline{\sigma}_i)} \right.$$

$$\left. + \sum_{n=L}^{U} \log P_{V_i}(\omega_n; \underline{\sigma}_i) \right\} \tag{158}$$

This algorithm is similar in structure and computation to the EM algorithm developed for the stationary (delay only) case. As before, the complicated multi-parameter optimization is decomposed into optimizations in smaller dimensional parameter subspaces, leading to a considerable simplification in the computations involved. As before, convergence is guaranteed to the set of stationary points of the likelihood function, with each iteration increasing the likelihood.

To solve (157), we must carry out a two-dimensional search in the delay-Doppler plane. We may consider replacing the maximization with the following coordinate search:

$$\hat{\tau}_i^{(l+1)} \leftarrow \max_{\tau_i} \left| \mathrm{Re}\left[\sum_{n=L}^{U} e^{-j(\omega_n - \widehat{\Delta\omega}_i^{(l)})\tau_i} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n - \widehat{\Delta\omega}_i^{(l)}) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})\right] \right|$$

$$\widehat{\Delta\omega}_i^{(l+1)} \leftarrow \max_{\Delta\omega_i} \left| \mathrm{Re}\left[\sum_{n=L}^{U} e^{-j(\omega_n - \Delta\omega_i)\hat{\tau}_i^{(l+1)}} \hat{S}^{(l)}(\omega_n) Z_i^*(\omega_n - \Delta\omega_i) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})\right] \right| \tag{159}$$

In this setting, the new delay estimate $\hat{\tau}_i^{(l+1)}$ is obtained by performing a weighted cross-correlation between the signal estimate and a Doppler shifted version of the signal at the i^{th} receiver output. The Doppler estimate $\widehat{\Delta\omega}_i^{(l+1)}$ is obtained by performing a frequency domain convolution between the weighted signal estimate $\hat{S}^{(l)}(\omega_n)/P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})$ and the i^{th} receiver output delayed in time by $\hat{\tau}_i^{(l+1)}$. Replacing the two-parameter optimization (157) by the indicated two-step procedure, we obtain a computationally attractive algorithm, which is guaranteed to converge to the set of stationary points of the likelihood function, and which increases the likelihood on each step. The disadvantage is that its convergence rate will be slower.

8.2 EM-ML Delay, Doppler Estimation

As indicated before, the problem with the EM algorithm is its slow linear rate of convergence. Therefore, it may be useful to develop hybrid EM-ML types of algorithms. To this end, suppose we want to estimate the delay and Doppler parameters by direct maximization of $L_Z(\underline{\xi})$, that is:

$$\hat{\underline{\tau}}^{(l+1)}, \widehat{\underline{\Delta\omega}}^{(l+1)} \leftarrow \max_{\underline{\tau},\underline{\Delta\omega}} L_Z(\underline{\tau}, \underline{\Delta\omega}, \hat{\underline{\alpha}}^{(l)}, \hat{\underline{\theta}}^{(l)}, \hat{\underline{\sigma}}^{(l)}) \tag{160}$$

where $\hat{\underline{\alpha}}^{(l)}$, $\hat{\underline{\theta}}^{(l)}$, and $\hat{\underline{\sigma}}^{(l)}$ are generated by the EM algorithm in the previous section.

Recall (151). By assumption (4), the term:

$$\sum_{n=L}^{U} \frac{|Z_i(\omega_n - \Delta\omega_i)|^2}{P_{V_i}(\omega_n;\underline{\sigma}_i)} = \sum_{n=L}^{U} \frac{|Z_i(\omega_n)|^2}{P_{V_i}(\omega_n - \Delta\omega_i;\underline{\sigma})} \approx \sum_{n=L}^{U} \frac{|Z_i(\omega_n)|^2}{P_{V_i}(\omega_n;\underline{\sigma}_i)} \tag{161}$$

is independent of $\Delta\omega_i$. Therefore, retaining only terms that depend significantly on $\underline{\tau}$ and $\underline{\Delta\omega}$:

$$\begin{aligned} L_Z(\underline{\xi}) &= c + \sum_{n=L}^{U} \frac{P_S(\omega_n;\underline{\theta})\left|\sum_{i=1}^{M} \alpha_i e^{-j(\omega_n-\Delta\omega_i)\tau_i} Z_i^*(\omega_n - \Delta\omega_i)/P_{V_i}(\omega_n;\underline{\sigma}_i)\right|^2}{1 + \sum_{i=1}^{M} \alpha_i^2 P_S(\omega_n;\underline{\theta})/P_{V_i}(\omega_n;\underline{\sigma}_i)} \\ &= c + \sum_{n=L}^{U}\sum_{i=1}^{M} \frac{\alpha_i^2 P_S(\omega_n;\underline{\theta})|Z_i(\omega_n - \Delta\omega_i)|^2/P_{V_i}^2(\omega_n;\underline{\sigma})}{1 + \sum_{i=1}^{M} \alpha_i^2 P_S(\omega_n;\underline{\theta})/P_{V_i}(\omega_n;\underline{\sigma}_i)} \\ &\quad + 2\mathrm{Re}\sum_{n=L}^{U}\sum_{i=1}^{M-1}\sum_{k=i+1}^{M} \frac{\alpha_i\alpha_k Z_i^*(\omega_n - \Delta\omega_i) Z_k(\omega_n - \Delta\omega_i) e^{j[(\omega_n-\Delta\omega_k)\tau_k - (\omega_n-\Delta\omega_i)\tau_i]}}{P_{V_i}(\omega_n;\underline{\sigma}_i)P_{V_k}(\omega_n;\underline{\sigma}_k)\left[1/P_S(\omega_n;\underline{\theta}) + \sum_{i=1}^{M}\alpha_i^2/P_{V_i}(\omega_n;\underline{\sigma}_i)\right]} \end{aligned} \tag{162}$$

where c is independent of $\underline{\tau}$ and $\underline{\Delta\omega}$. If $P_S(\omega_n;\underline{\theta})$ and $P_{V_i}(\omega_n;\underline{\sigma}_i)$ are smooth over the signal bandwidth, and the maximum Doppler shift is small compared to the signal bandwidth, then the first term in (162) does not depend strongly on $\Delta\omega_i$ and can be ignored. This leaves:

$$\begin{aligned} \hat{\underline{\tau}}^{(l+1)}, \widehat{\underline{\Delta\omega}}^{(l+1)} \leftarrow \max_{\underline{\tau},\underline{\Delta\omega}} \mathrm{Re}\Bigg[&\sum_{n=L}^{U}\sum_{i=1}^{M-1}\sum_{k=i+1}^{M} \hat{W}_{ik}^{(l)}(\omega_n) Z_i^*(\omega_n - \Delta\omega_i) Z_k(\omega_n - \Delta\omega_i) \\ &* e^{j[(\omega_n-\Delta\omega_k)\tau_k - (\omega_n-\Delta\omega_i)\tau_i]}\Bigg] \end{aligned} \tag{163}$$

where $\hat{W}_{ik}^{(l)}(\omega_n)$ is the same weighting function that was defined in (75). It can be shown that for $WT/2\pi \gg 1$, the solution to (163) converges in about one step (super-linearly) to the true $\underline{\tau}$ and $\underline{\Delta\omega}$, regardless of the available gain and spectral estimates.

However, it requires a complicated multi-dimensional search over all possible delays and Dopplers.

We may consider maximizing $L_Z(\underline{\xi})$ with respect to one pair of $(\tau_i, \Delta\omega_i)$ at a time. Similarly to (113), we obtain the following algorithm:

$$\text{For } l = 0, 1, \ldots$$

$$\hat{\tau}_i^{(l+1)}, \widehat{\Delta\omega}_i^{(l+1)} \leftarrow \max_{\tau_i, \Delta\omega_i} \text{Re} \sum_{n=L}^{U} \frac{e^{-j(\omega_n - \Delta\omega_i)\tau_i} \hat{S}_i^{(l)}(\omega_n) Z_i^*(\omega_n - \Delta\omega_i)}{P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)})} \tag{164}$$

where:

$$\hat{S}_i^{(l)}(\omega_n) = \frac{\sum_{\substack{k=1 \\ k \neq i}}^{M} P_S(\omega_n; \underline{\hat{\theta}}^{(l)}) e^{j(\omega_n - \widehat{\Delta\omega}_k)\hat{\tau}_k} Z_k(\omega_n - \widehat{\Delta\omega}_k) / P_{V_k}(\omega_n; \underline{\hat{\sigma}}_k^{(l)})}{1 + \sum_{k=1}^{M} \widehat{SNR}_k^{(l)}(\omega_n)} \tag{165}$$

where:

$$\hat{\tau}_k = \begin{cases} \hat{\tau}_k^{(l+1)} & \text{for } k < i \\ \hat{\tau}_k^{(l)} & \text{for } k > i \end{cases} \tag{166}$$

$$\widehat{\Delta\omega}_k = \begin{cases} \widehat{\Delta\omega}_k^{(l+1)} & \text{for } k < i \\ \widehat{\Delta\omega}_k^{(l)} & \text{for } k > i \end{cases} \tag{167}$$

Equation (164) is similar to (157), except that in (164) we use $\hat{S}_i^{(l)}(\omega_n)$, the signal estimate computed using all but the i^{th} receiver outputs. This algorithm represents a compromise between the EM algorithms and the EM-ML algorithm above. It requires slightly more effort than the EM algorithm, since we need to generate $\hat{S}_i^{(l)}(\omega_n)$ for all $i = 1, \ldots, M$. However, we may expect significantly better convergence behavior. When compared with (163), it replaces the multi-parameter optimization with a coordinate search, thus trading off computational complexity for a slower rate of convergence.

We can further simplify the computations per iteration at the cost of reducing the rate of convergence, by performing a coordinate search with respect to each delay and Doppler parameter separately. We obtain:

$$\begin{aligned} \hat{\tau}_i^{(l+1)} &\leftarrow \max_{\tau_i} \left| \text{Re} \left[\sum_{n=L}^{U} e^{-j(\omega_n - \widehat{\Delta\omega}_i^{(l)})\tau_i} \hat{S}_i^{(l)}(\omega_n) Z_i^*(\omega_n - \widehat{\Delta\omega}_i^{(l)}) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)}) \right] \right| \\ \widehat{\Delta\omega}_i^{(l+1)} &\leftarrow \max_{\Delta\omega_i} \left| \text{Re} \left[\sum_{n=L}^{U} e^{-j(\omega_n - \Delta\omega_i)\hat{\tau}_i^{(l+1)}} \hat{S}_i^{(l)}(\omega_n) Z_i^*(\omega_n - \Delta\omega_i) / P_{V_i}(\omega_n; \underline{\hat{\sigma}}_i^{(l)}) \right] \right| \end{aligned} \tag{168}$$

This algorithm should converge more quickly than the EM algorithm (159). A convergence analysis could be carried out using the same methods as in section 5 to quantify the trade-off between the computational burden and convergence rate.

8.3 Other EM-ML Algorithms

In order to accelerate the convergence of the signal gains and spectral parameters, we may try a hybrid EM-ML algorithm which directly maximizes the log-likelihood over $\underline{\alpha}$ and $\underline{\sigma}$, and only uses the EM step to update the $\underline{\theta}$ parameters:

$$\underline{\hat{\alpha}}^{(l+1)}, \underline{\hat{\sigma}}^{(l+1)} \leftarrow \max_{\underline{\alpha},\underline{\sigma}} L_Z\left(\underline{\hat{\tau}}^{(l+1)}, \underline{\widehat{\Delta\omega}}^{(l+1)}, \underline{\alpha}, \underline{\sigma}, \underline{\hat{\theta}}^{(l)}\right) \tag{169}$$

$$\underline{\hat{\theta}}^{(l+1)} \leftarrow \max_{\underline{\theta}} \sum_{n=L}^{U} \left[\log P_S(\omega_n;\underline{\theta}) + \frac{|\widehat{S(\omega_n)}|^{2^{(l)}}}{P_S(\omega_n;\underline{\theta})}\right]$$

We may even try to replace the signal spectral estimate with a direct likelihood maximization:

$$\underline{\hat{\theta}}^{(l+1)} \leftarrow \max_{\underline{\theta}} L_Z\left(\underline{\hat{\tau}}^{(l+1)}, \underline{\widehat{\Delta\omega}}^{(l+1)}, \underline{\hat{\alpha}}^{(l+1)}, \underline{\hat{\sigma}}^{(l+1)}, \underline{\theta}\right) \tag{170}$$

and not use the EM algorithm at all. Note, however, that the log-likelihood for the Doppler case (151) is identical to that for the stationary case (13) except that terms $Z_i(\omega_n)$ are replaced by $Z_i(\omega_n - \Delta\omega_i)e^{-j\Delta\omega_i\tau_i}$. Otherwise, the likelihoods have exactly the same dependence on $\underline{\alpha}$, $\underline{\theta}$, and $\underline{\sigma}$. Therefore, all the considerations and results developed in section 5 are directly applicable here by simply substituting:

$$Z_i(\omega_n) \rightarrow Z_i(\omega_n - \widehat{\Delta\omega}_i^{(l+1)})e^{-j\widehat{\Delta\omega}_i^{(l+1)}\hat{\tau}_i^{(l+1)}} \tag{171}$$

where $\hat{\tau}_i^{(l+1)}$ and $\widehat{\Delta\omega}_i^{(l+1)}$ denote the most recent estimates of τ_i and $\Delta\omega_i$ respectively.

9 Conclusions

In this report, we have developed a new class of iterative algorithms for solving the Maximum Likelihood time delay estimation problem given multiple receivers, with unknown signal, unknown signal power spectrum, unknown noise power spectra, and unknown signal gains at each receiver. Our algorithms all decouple the various unknown parameters, estimating only one unknown at a time, thereby reducing a difficult nonlinear multidimensional optimization to a series of simpler, lower dimensional optimizations. Furthermore, the methods naturally generate an estimate of the signal waveform, which can be used for target identification or other purposes. Each iteration increases the likelihood function, and convergence is guaranteed to the set of stationary points of the likelihood function.

We've suggested four different classes of iterative time-delay estimation algorithms. The EM-like algorithm estimates the signal from a weighted, Weiner-Hopf filtered combination of receiver data. It then cross-correlates this signal estimate against each receiver to estimate the time delay, and the peak value of the cross-correlation is normalized to form the signal gain. The noise spectrum model is fit to the remaining error on each channel, and the signal power spectrum model is fit to the estimated signal periodogram. All-pole models are particularly easy to use. The method then iterates, using the improved spectral models to build better filters for the signal estimate. The computation required is thus quite simple. The method has difficulty, however, with initial convergence due to an extra peak at the old estimate in the objective function being maximized for each delay. The asymptotic convergence rate can also be slow when the SNR is low, and when there are few channels. In cases with large numbers of channels or high SNR, however, the delay estimates will converge in about 2 iterations, and the normalized signal gains and power spectra will converge in about 5 iterations.

Better convergence is achieved by the EM-ML delay estimation algorithm. This approach directly maximizes the likelihood function to estimate the delays. If all delays are estimated simultaneously, then superlinear convergence of the delay estimates can be achieved. If the delays are estimated one by one, the computation is much

simpler, but the convergence rate is slower. The i^{th} delay is estimated by maximizing a cross-correlation between a signal estimate and the i^{th} receiver data, where the signal is estimated using all received data except from the i^{th} channel. The asymptotic convergence rate is approximately independent of the SNR or the number of channels, and is between 1/8 and 1/9. In practice, about 2-3 iterations are needed to estimate the delays, and about 3-4 iterations to estimate the signal gains.

Somewhat better convergence of the gain estimates can be achieved by modifying this EM-ML algorithm so that the signal gains are also estimated by directly maximizing the likelihood function. This EM-ML delay/gain method can only estimate a single noise level for all channels. To compute the new gains, we find the maximum eigenvalue and corresponding eigenvector of a weighted correlation matrix formed from the receiver data. The gains are the eigenvector components, and the noise level is derived from the eigenvalue. In theory, superlinear convergence of the gains can be achieved if the delays are estimated jointly. In practice, if we estimate the delays one at a time, and estimate the noise level as well as the signal power spectrum, convergence of the gains is only accelerated for SNR near threshold (about 0dB). At high SNR, the improvement in the convergence rate is probably not striking enough to justify solving an eigenvector problem.

Finally, we suggested a direct coordinate ascent procedure for directly maximizing the likelihood for all the parameters. To do this, we must restrict ourselves to a single noise level parameter for all channels, and can only estimate a non-parameterized signal power spectrum model. The resulting algorithm is identical to the EM-ML delay/gain method, but rather than fitting an all-pole model for the signal spectrum, it uses an expected periodogram estimate. Convergence of this direct ML method is similar to that of the EM-ML delay/gain method. This observation suggests that getting excellent models for the signal power spectrum is not essential for getting good delay or gain estimates. Unfortunately, because the method requires solving an eigenvector problem for the gains, it is computationally unattractive.

From our work in this paper, we conclude that the EM algorithm and the EM-ML delay estimate algorithm are computationally simple and very attractive algorithms

for getting highly accurate time delay estimates at an array of receivers. Unlike the generalized cross-correlation (GCC) method, they do not require a priori estimates of the signal or noise spectra or the signal gains, since they estimate these as they iterate. Despite having to estimate so many parameters, the EM and EM-ML delay estimation algorithms achieve delay estimates whose variance is close to that of the GCC method, and is better than the simple cross-correlation method. With little effort at providing good initial guesses, we were able to obtain reliable convergence in 1-5 iterations for between 2 to 8 channels, and for a variety of SNR above threshold. Since the primary advantage of the likelihood-based methods (and the GCC method as well) is that they incorporate filtering, maximum advantage is obtained in examples where the signal is narrow-band compared to the noise.

Many issues remain to be explored. The analysis techniques used in this paper are quite novel, and have important implications for the analysis of nearly all EM-style algorithms, in applications ranging from iterative clustering to adaptive equalization. It should be possible, using these techniques, to derive analytical formulas for the convergence rate of nearly all these algorithms. Furthermore, our general iterative framework allows us to build hybrid iterative algorithms for a very wide range of applications. In general, these algorithms should enjoy faster convergence rates than the conventional EM approach, although at the possible cost of increased computation.

Another important area of research involves extensions of these algorithms for related applications. The most important extension is to estimate multiple signals received at an array of sensors from different directions. The formulas for this case do not decouple quite as nicely as the single source case, because of the non-orthogonality of the received signal wavefronts. However, we expect that the general structure of this iterative multiple source estimator would use multiple beamformers, each adapted to preferentially pass only one of the signals, with iteration being used to model the signal, model the noise, and adapt the beam weights and delays. Other important extensions are to use pole-zero models, and to consider more complicated parameterized signal and noise models.

Another useful extension would be to consider recursive versions of these algorithms.

A convenient framework for this would be within the context of linear state space models. A technical difficulty is that we need to use a continuous-time model for the signal, in order to allow arbitrary receiver delays, but must use a discrete-time model for the observations. We are currently working on extending our batch algorithms to this recursive model. Our ultimate goal would be a Kalman Filter-like interpolator/filter for estimating the signal, with a recursive update for the signal and noise model parameters. The major technical difficulty we expect is that if the delay estimates should get "lost", it is not clear whether the algorithm will "recapture" the correct delay values.

To conclude, the research area of iterative EM and EM-like algorithms has great importance for a variety of stochastic estimation problems. It not only yields new insight into the estimation process itself, but also suggests many new and practical estimation algorithms.

A EM-ML Gain and Noise Level Estimates

In this appendix, we will maximize equation (119) over $\underline{\alpha}$ and σ to get the EM-ML gain and noise level estimates. Note that (119) is quadratic in the signal gains $\underline{\alpha}$. To maximize it subject to the constraint (118), introduce a Lagrange multiplier λ and form the Lagrangian:

$$\mathcal{L}(\xi) = -\sum_{n=L}^{U} M \log\sigma - \frac{1}{\sigma}\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)} + \frac{N}{\sigma^2}\underline{\alpha}^T\mathbf{R}(\underline{\tau},\underline{\theta})\underline{\alpha} - \lambda\left(\frac{1}{\gamma\sigma}\sum_{i=1}^{M}\alpha_i^2 - 1\right) \quad (172)$$

The maximum of $L_Z(\xi)$ over $\underline{\alpha}$ and σ subject to (119) can then be found by computing the derivatives of $\mathcal{L}(\xi)$ with respect to $\underline{\alpha}$ and σ and setting them to zero. This gives:

$$\frac{2N}{\sigma^2}\mathbf{R}(\underline{\tau},\underline{\theta})\underline{\alpha} - \frac{2\lambda}{\gamma\sigma}\underline{\alpha} = 0 \quad (173)$$

$$\frac{1}{\gamma\sigma}\sum_{i=1}^{M}\alpha_i^2 - 1 = 0 \quad (174)$$

$$-\frac{MN}{\sigma} + \frac{1}{\sigma^2}\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)} - \frac{2N}{\sigma^3}\underline{\alpha}^T\mathbf{R}(\underline{\tau},\underline{\theta})\underline{\alpha} + \frac{\lambda}{\gamma\sigma^2}\sum_{i=1}^{M}\alpha_i^2 = 0 \quad (175)$$

Combining these expressions, setting $\mu = \lambda\sigma/N\gamma$, and applying much algebra gives the simplified form:

$$\mathbf{R}(\underline{\tau},\underline{\theta})\underline{\alpha} = \mu\underline{\alpha} \quad (176)$$

$$\sigma = \frac{1}{MN}\sum_{n=L}^{U}\sum_{i=1}^{M}\frac{|Z_i(\omega_n)|^2}{P_V(\omega_n)} - \mu\gamma/M \quad (177)$$

$$\frac{1}{\gamma\sigma}\underline{\alpha}^T\underline{\alpha} = 1 \quad (178)$$

Thus $\underline{\alpha}$ must be an eigenvector of $\mathbf{R}$, and μ is the corresponding eigenvalue. Which eigenvalue and eigenvector? Substituting back into (119),

$$L_Z(\xi) = c' - MN\log\sigma \quad (179)$$

where c' is a constant. Clearly to maximize the log likelihood we should choose μ such that σ is as small as possible. Thus μ should be chosen as the largest eigenvalue of $\mathbf{R}$, and $\underline{\alpha}$ is the corresponding eigenvector.

B EM-ML Individual Delay Estimation Algorithm - Asymptotic Convergence Rate

In this appendix we analyze the asymptotic convergence rate of the EM-ML individual delay estimation algorithm. Assume that all channels have equal signal and noise energy, so that all functions $\hat{R}_{s_{ik}}^{(l)}(t)$ can be approximated with the same Taylor series approximation (86), with $\beta_{ik}^{(l)} = \beta^{(l)}$. Substituting (86) into (116) and setting the derivatives with respect to τ_i to zero,

$$\hat{\tau}_i^{(l+1)} \approx \bar{\tau}_i + \eta \left[\sum_{k=1}^{i-1} \left(\hat{\tau}_k^{(l+1)} - \bar{\tau}_k \right) + \sum_{k=i+1}^{M} \left(\hat{\tau}_k^{(l+1)} - \bar{\tau}_k \right) \right] \tag{180}$$

where $\eta = 1/(M-1)$. To analyze the convergence rate, define a state vector $\underline{\tilde{\tau}}[n]$ by cyclically permuting the delay estimates:

$$\underline{\tilde{\tau}}[i + M(l+1)] = \begin{pmatrix} \hat{\tau}_{i+1}^{(l)} - \bar{\tau}_{i+1} \\ \vdots \\ \hat{\tau}_M^{(l)} - \bar{\tau}_M \\ \hat{\tau}_1^{(l+1)} - \bar{\tau}_1 \\ \vdots \\ \hat{\tau}_i^{(l+1)} - \bar{\tau}_i \end{pmatrix} \tag{181}$$

Then estimating a single delay is equivalent to:

$$\underline{\tilde{\tau}}[n+1] = \Phi \underline{\tilde{\tau}}[n] \tag{182}$$

where Φ is an $M \times M$ matrix which is in companion form, and is independent of i or l:

$$\Phi = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & \eta & \eta & \cdots & \eta \end{pmatrix} \tag{183}$$

Estimating all M delays on one iteration is equivalent to applying the Φ matrix M times to the previous delay estimates. Thus:

$$\begin{pmatrix} \hat{\tau}_1^{(l+1)} - \bar{\tau}_1 \\ \vdots \hat{\tau}_M^{(l+1)} - \bar{\tau}_M \end{pmatrix} = \Phi^M \begin{pmatrix} \hat{\tau}_1^{(l)} - \bar{\tau}_1 \\ \vdots \hat{\tau}_M^{(l)} - \bar{\tau}_M \end{pmatrix} \tag{184}$$

The rate of convergence depends on the eigenvalues of Φ. To find these, compute the characteristic polynomial of Φ:

$$\begin{aligned} \det(sI - \Phi) &= s^M - \eta s^{M-1} - \eta s^{M-2} - \cdots - \eta s \\ &= s(s-1)\left((M-1)s^{M-2} + (M-2)s^{M-3} + \cdots + 2s + 1\right)/(M - \quad (185) \\ &= s\frac{1 - Ms^{M-1} + s^M}{1 - s} \end{aligned} \tag{186}$$

Φ has one eigenvalue of 1. This corresponds to our decision to estimate all M delays, despite the fact that only the $M-1$ relative delays can be identified from the data. Therefore the average level of the delay estimates cannot be identified, and any constant delay error will remain unchanged. The eigenvector corresponding to eigenvalue 1 is exactly this constant error vector. The convergence rate of the relative delays is therefore equal to the absolute value of the root of the polynomial on the right of (185) with the largest magnitude, raised to the M^{th} power. Figure 21 plots the convergence rate for M in the range 3 through 150. Note that the convergence rate is between about 1/8 and 1/9, independent of SNR and independent of the number of channels M.

For unequal signal energy in the channels, we would expect faster convergence. For example, if channel M is much stronger than the others, then its contribution to the likelihood function will overwhelm the other channels, and each delay estimate will tend to set $\hat{\tau}_i^{(l+1)} - \hat{\tau}_M^{(l+1)}$ to about the right value.

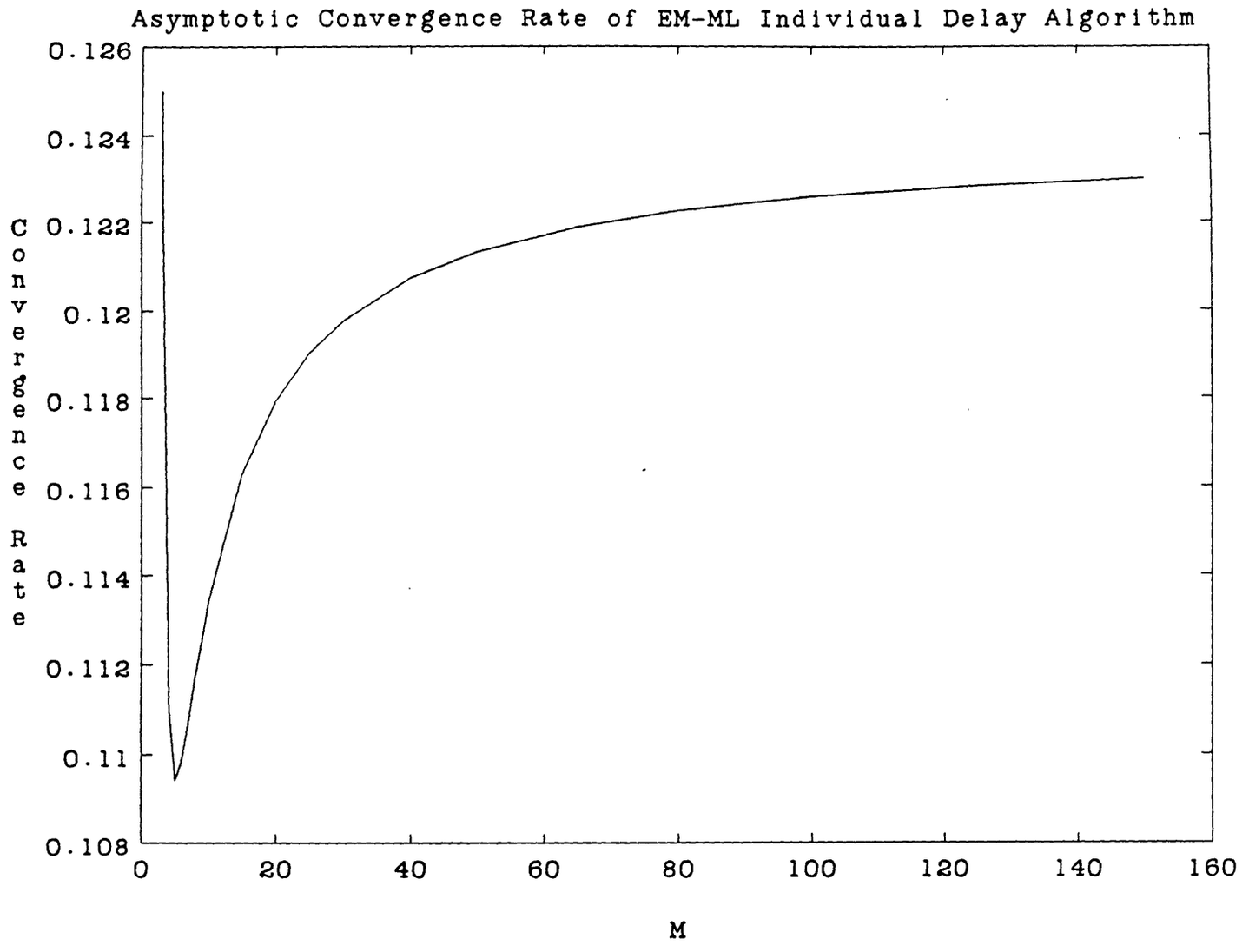


Figure 21: Asymptotic convergence rate of EM-ML individual delay algorithm

References

[1] Special issue on time delay estimation. IEEE Trans. Acoust., Speech, and Signal Processing, June 1981.

[2] B. Anderson and J. Moore. *Optimal Filtering.* Volume of , Prentice Hall, New Jersey, edition, 1979.

[3] G. Carter, A. Nuttall, and P. Cable. The smoothed coherence transform. *Proc. IEEE (Lett.)*, 61():1497–1498, Oct. 1973.

[4] Y. Chan, J. Riley, and J. Plant. A parameter estimation approach to time delay estimation and signal detection. *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-28(1):8–16, Feb. 1980.

[5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. Ser. B*, 39():1–38, 1977.

[6] C. Eckart. *Optimal Rectifier Systems for the Detection of Steady Signals.* SIO 12692, SIO Ref 52-11, Univ. California, Scripps Inst. Oceanography, Marine Physical Lab, 1952.

[7] B. Hamon and E. Hannan. Spectral estimation of time delay for dispersive and non-dispersive systems. *J. Roy. Stat. Soc. Ser. C. (Appl. Statis.)*, 23(2):134–142, 1974.

[8] W. Hann. Optimum signal processing for passive sonar range and bearing estimation. *J. Acoust. Soc. Amer.*, 58(2):201–207, 1975.

[9] E. Hannan and P. Thomson. Delay estimation and the estimation of coherence and phase. *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-29(3):485–490, June 1981.

[10] E. Hannan and P. Thomson. Estimating group delay. *Biometrika*, 60():241–253, 1973.

[11] C. Knapp and G. Carter. Estimation of time delay in the presence of source or receiver motion. *J. Acoust. Soc. Amer.*, 61(6):1545–1549, June 1977.

[12] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-24(4):320–327, August 1976.

[13] V. MacDonald and P. Schultheiss. Optimum passive bearing estimation in a spatially incoherent noise enviroment. *J. Acoust. Soc. Amer.*, 46(1 (Pt. 1)):37–43, 1969.

[14] I. Meilijson. A fast improvement to the EM algorithm on its own terms. to be published in J. Roy. Stat. Soc. Ser. B, .

[15] B. Musicus. *Iterative Algorithms for Optimal Signal Reconstruction and Parameter Identification Given Noisy and Incomplete Data.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Sept. 1982.

[16] B. Musicus and J. Lim. Maximum likelihood parameter estimation of noisy data. In , editor, *Proc. 1979 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 224–227, IEEE, Washington, D.C., April 1979.

[17] P. Roth. Effective measurements using digital signal analysis. *IEEE Spectrum*, 8():62–70, April 1971.

[18] P. Schultheiss and E. Weinstein. Estimation of differential doppler shifts. *J. Acoust. Soc. Amer.*, 55(5):1412–1419, Nov. 1979.

[19] E. Weinstein. *Estimation of Trajectory Parameters from Passive Array Measurements.* PhD thesis, Yale Univ., New Haven, CT, 1978.

[20] C. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, (11):95–103, 1983.