

Research Article

Modified Particle Swarm Optimization for Blind Deconvolution and Identification of Multichannel FIR Filters

Vahid Khanagha,¹ Ali Khanagha,² and Vahid Tabataba Vakili¹

¹Electrical Engineering Department, Iran University of Science & Technology (IUST), Narmak, 1684613114 Tehran, Iran

²Research Institute of Petroleum Industry (RIPI), Shahre Rey, 1485733111 Tehran, Iran

Correspondence should be addressed to Vahid Khanagha, vahid.khanagha@inria.fr

Received 22 March 2009; Revised 8 December 2009; Accepted 13 January 2010

Academic Editor: Irene Gu

Copyright © 2010 Vahid Khanagha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blind identification of MIMO FIR systems has widely received attentions in various fields of wireless data communications. Here, we use Particle Swarm Optimization (PSO) as the update mechanism of the well-known inverse filtering approach and we show its good performance compared to original method. Specially, the proposed method is shown to be more robust against lower SNR scenarios or in cases with smaller lengths of available data records. Also, a modified version of PSO is presented which further improves the robustness and preciseness of PSO algorithm. However the most important promise of the modified version is its drastically faster convergence compared to standard implementation of PSO.

1. Introduction

This paper addresses the problem of blind identification of multi-input-multi-output (MIMO) channels in the general scenario, where all of the M observed signals $y(n) = [y_1(n), \dots, y_M(n)]^T$ are considered to be convolutive mixtures of N unknown, yet statistically independent sources $x(n) = [x_1(n), \dots, x_N(n)]^T$. This is often the case in typical echoic environments, where each sensor captures not only the direct contributions from each source, but also several reflected versions of the original signals with totally different propagation delays. The blindness here means that we have no prior knowledge about the MIMO system and no training data is available. The i th observed mixture can be written as

$$y_i(k) = \sum_{j=1}^M \sum_{l=-\infty}^{\infty} f_{ij}(k-l)x_j(l) + n_i(k). \quad (1)$$

Here $n_i(k)$ is additive white Gaussian noise and $f_{ij}(l)$, $-\infty < l < \infty$ are tap weight coefficients of channel impulse response from j th source to the i th sensor. Each subchannel of the MIMO system can be written in z domain as $F_{ij}(z) = \sum_{l=-\infty}^{\infty} f_{ij}(l)z^{-l}$.

Inverse filtering approach [1] consists of an iterative solution, which recursively extracts sources from the mixture one by one and then following the estimation of experienced channel by any extracted source, and reconstructs this signal as it was originally observed on each sensor. Now, after reduction of reconstructed sources from sensor measurements, the same procedure can be used for extraction of remaining sources. The source extraction step is down by the steepest descent maximization of a class of cumulant-based cost functions with respect to coefficients of equalizer filters. However, this optimization strategy is prone to being trapped in local maxima, especially in lower SNR scenarios or when the available data record is too small [1].

An alternative to this gradient-based optimization is a structured stochastic search of the objective function space. These types of global searches are structure independent because a gradient is not calculated and the adaptive filter structure does not directly influence the parameter updates. Due to this property these types of algorithms are potentially capable of globally optimizing any class of objective functions [2]. Particle Swarm Optimization (PSO) is one of these stochastic structured search algorithms that have recently gained popularity for optimization problems.

This paper investigates the application of PSO technique in source extraction step of the above-mentioned procedure for mutually independent, zero mean i.i.d binary sequences. It should be noted that although [3] has addressed the same subject, but its update equation seems more like a heuristic method which employs a random weighted addition of gradients based on second-order statistics and prediction methods, while PSO is defined as cooperative random search of particles toward their current global and local best points in search space according to some fitness function [4] as it will be discussed in Section 3.1 and simple multiplication of update equation by a random number (as in [3]) does not represent the essence of global random search suggested by PSO.

In this paper, after studying the suitability of standard PSO for blind identification problem, we propose a modified version which finds its initial direction according to original gradient-based method in [1]. In fact the promised random search of PSO will be limited to smaller local areas with the most probability of maximizing the cost function. Also we modify the original fitness function used in [1, 3] by two supportive performance indexes in order to reduce the probability of algorithm failures which are the result of complete ignorance of original objective function about the existence of additive noise.

2. Iterative Source Extraction and Channel Estimation

In this section, we provide a detailed description of inverse filtering approach. Note that here we neglect the presence of noise $n_i(k)$ in (1). Consider that a $1 \times N$ row vector \mathbf{C}^T , with its i th element $C_i(z) = \sum_{l=1}^L c_i(l)z^{-l}$, operates on observed data vector $\mathbf{y}(k)$ to construct equalizer output as

$$\begin{aligned} e(k) &= \sum_{i=1}^N y_i(k) * c_i(k) \\ &= \sum_{i=1}^N \sum_{j=1}^M x_j(k) * f_{ij}(k) * c_i(k) \\ &= \sum_{i=1}^N \sum_{j=1}^M x_j(k) * h_j(k). \end{aligned} \quad (2)$$

Here $h_j(l)$, $j = -\infty, \dots, \infty$, are elements of the overall impulse response from j th source to equalizer output:

$$H_j(z) = F_{ij}(z)C_i(z) = \sum_{l=-\infty}^{\infty} h_j(l)z^{-l}. \quad (3)$$

Based on third- and fourth- order cumulant of equalizer output, [1] introduced a set of objective functions J_{r2} as

$$J_{r2} = \frac{|\text{CUM}_r(e(k))|}{|\text{CUM}_2(e(k))|^{r/2}}. \quad (4)$$

Here $\text{CUM}_k(e(k))$ is k th order cumulant of the equalizer output. The following upper bound has derived in [1] for J_{r2} :

$$J_{r2}(c) \leq \left| \bar{y}_{r \max} \right|. \quad (5)$$

Here, $|\bar{y}_{r \max}| = \max_{1 \leq j \leq M} |\text{CUM}_r(x_j(k))|$. Equality of (5) holds when the overall transfer function becomes

$$h_j(k) = d\delta(j - j_0)\delta(k - k_0). \quad (6)$$

So the maximization of J_{r2} with respect to equalizer coefficients \mathbf{C}^T leads to extraction of one of the sources on equalizer output up to a scaled and delayed version as

$$e(k) = \alpha x_{j_0}(k - k_0), \quad j_0 \in \{1, \dots, M\}. \quad (7)$$

This separation criteria, however, has the weakness of complete ignorance about the presence of noise. Authors in [1] have developed their theoretical proof about suitability of the above-mentioned objective function based on the assumption that the noise $n(t)$ in (1) is negligible. This is the reason for poor performance of their proposed iterative, batch, steepest descent method f optimization at lower SNRs.

If we consider the presence of noise $n_i(k)$ in (1), the addition of the term $\sum_{i=1}^N n_i(k) * c_i(k)$ to $e(k)$ of (2) will destruct the validity of the theoretical proof in [1]. In fact when noise is dominant in sensor measurements, cooperation of this noise with equalizer coefficients may build up new fake maxima into J_{r2} regardless of overall impulse response $h_j(k)$ and in this new maximum, separation point of (6) is no longer achievable.

However it is obvious that (6) would still provide local maximum for J_{r2} and separation would be met if we could somehow control the algorithm from trapping into these new maximum as it will be explained in Section 3.2.

After extraction of each source, the estimated signal can be used for estimation of the experienced channel by that signal through its way to each sensor:

$$\hat{f}_{i,j_0}(\tau) := \frac{E\{y_i(k)e^*(k - \tau)\}}{E\{|e(k)|^2\}}. \quad (8)$$

Now, the third step is to reconstruct the extracted source as it was originally observed on each sensor in order to suppress its contribution from sensor observations:

$$\hat{y}_{i,j_0}(k) = \sum_l \hat{f}_{i,j_0}(l)e(k - l), \quad (9)$$

$$y'_i(k) = y_i(k) - \hat{y}_{i,j_0}(k), \quad y_i(k) \leftarrow y'_i(k).$$

Hereafter, the same procedure can be used for extraction of remaining sources and then estimating the SIMO channel experienced by any one of them.

3. Particle Swarm Optimization

3.1. PSO Principles. Particle swarm optimization [2] is a stochastic, population-based evolutionary algorithm for problem solving in complex multidimensional parameter spaces. It is a kind of swarm intelligence that is based on social-psychological principles.

A multidimensional optimization problem is given, along with an objective function to evaluate the fitness of each candidate point in parameter space; the swarm is typically modeled by particles in this multidimensional space that have a position and a velocity. After the definition of a random population of individuals (particles) as candidate solutions, they fly through hyperspace of parameters with the aid of two essential reasoning capabilities: their memory of their own best local position and knowledge of the global or their neighborhood's best [5].

PSO begins by initializing a random swarm of N_p particles $p_i = [p_{i1}, \dots, p_{iL}]$, each having L parameters. At each iteration, the fitness of each particle is evaluated according to selected fitness function. The fittest experienced position of each particle is stored and progressively replaced as $pbest_i, i = 1, \dots, N_p$, along with a single most globally fit particle ($gbest$) as fitter locations are encountered during algorithm iterations. The parameters of each particle in the swarm are updated at each iteration (n) according to a velocity vector as [6]

$$\begin{aligned} \overline{vel}_i(n) &= \omega \times \overline{vel}_i(n-1) \\ &+ acc_1 \times \text{diag}[e_1, e_2, \dots, e_L] \\ &\times (gbest - p_i(n-1)) \end{aligned} \quad (10)$$

$$\begin{aligned} &+ acc_2 \times \text{diag}[e_1, e_2, \dots, e_L] \\ &\times (pbest_i - p_i(n-1)), \\ p_i(n) &= p_i(n-1) + \overline{vel}_i(n). \end{aligned} \quad (11)$$

Here, $\overline{vel}_i(n)$ is the velocity vector of particle i , $e_r \in (0, 1)$ is a random value, acc_1 and acc_2 are acceleration coefficients toward $gbest$ and $pbest_i$, and ω is the inertia weight.

In fact, the trajectory of each particle is determined by random superposition of its previous velocity with the location of local and global best particles found by far. As new $gbests$ are encountered during the update process, all other particles begin to swarm toward this new $gbest$, continuing their random search along the way. The optimization is terminated when all of the particles have converged to $gbest$ or a sufficient condition of fitness function met:

3.2. Implementation of PSO for Source Extraction. We propose to use PSO as optimization method for maximization of J_{r2} with respect to coefficients of parallel equalizers ($c_i(l)$, $i = 1, \dots, M, l = 1, \dots, L$) as $M \times L$ dimensional particles.

Clearly J_{r2} of (4) seems a reasonable choice of fitness function since in the absence of noise, its maximization provides pure separation of the source $x_{j0}(k)$ at multichannel equalizer output. But as it is mentioned in Section 2 this

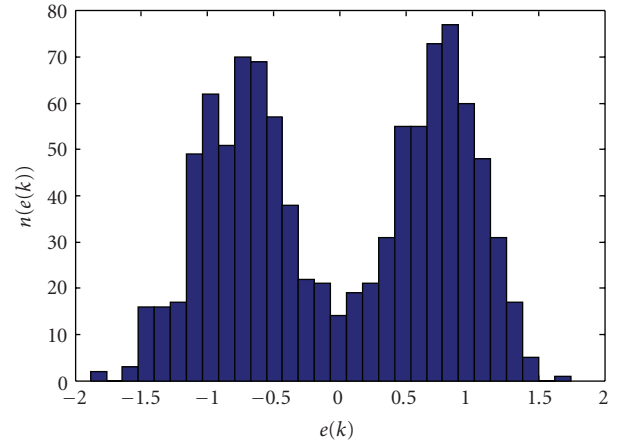


FIGURE 1: Histogram of successfully extracted source at equalizer's output.

objective function fails in low SNR scenarios and would have several fake maxima with respect to equalizer coefficients. Also the steepest gradient method of [1] has shown poor performance in cases with limited number of available data samples.

The simplicity of PSO suggests that we can easily combine several objective functions for further evaluation of the true level of fitness of a candidate particle in order to stay aside from the trap of fake global maxima in the case of noisy and smaller length records of data. For instance, since we expect complete deconvolution of extracted source at each iteration, a simple choice is to evaluate the level of correlation between successive samples of equalizer output. It is clear that there will be no time dependency between successive samples at ideal point of separation and deconvolution as in (5). It allows us to exploit signal correlations at nonzero lags as in [7]. Specifically we expect the following cost function to converge to zero when the true separation met.

$$J_{cor} = \sum_{l=1}^K E \left\{ [e(k)e(k+l)]^2 \right\}. \quad (12)$$

Also we can use histogram of equalizer output as a clue for leading particles toward desired solution. For the assumed equiprobable sequence of ∓ 1 s we expect the histogram of the equalized output to have a form like Figure 1 which shows the histogram of one of the successfully extracted sources in high SNR scenario. In lower SNR cases, when algorithms fails this histogram finds irregular Gaussian like shapes as in Figure 2. Two important differences between these two histograms exist: first, the concentration of histogram in Figure 2 around zero and second, the existence of infrequent extreme deviations as few points which are larger than 2.

Then as another measure of fitness, we can use the frequency of occurrence of samples with very small absolute values (smaller than 0.2 e.g.,) or with values larger than 2 as fitness parameter J_{hist} . Here by frequency of occurrence we

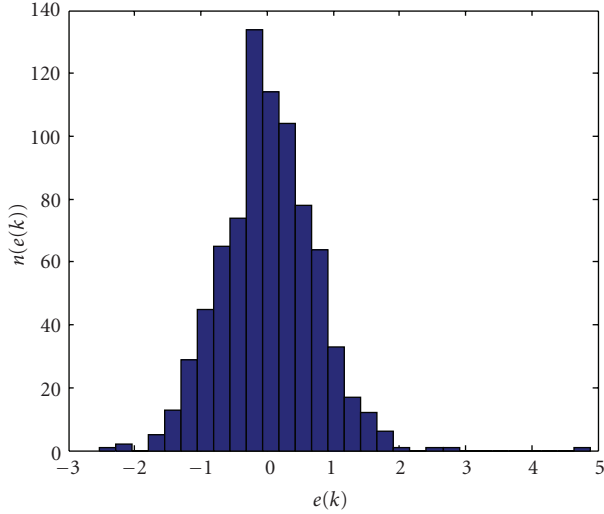


FIGURE 2: Irregular histogram of equalizer's output when algorithm fails.

mean the number of occurrences. Now, our overall fitness function of particles \mathbf{c}_i is defined as

$$J_{\text{fitness}}(\mathbf{c}_i) = \alpha J_{r2}(\mathbf{c}_i) - \beta J_{\text{cor}}(\mathbf{c}_i) - J_{\text{hist}}(\mathbf{c}_i). \quad (13)$$

Here α , β are approximately small coefficients and their selection has some effects on the convergence speed and robustness of the algorithms. In this new fitness function, J_{r2} and J_{cor} have comparable values as their weight should be adjusted by α and β . J_{hist} has very large modulus values which is desirable because it should strictly reject any particle with undesired histogram shape in the tails. Note that we can not apply this objective function to original gradient based method since the disapproval of a new equalizer coefficient set is equal to termination of algorithm run and it may causes the algorithm to stock near the initialization point, even in high SNR scenarios.

As it will be shown in Section 4, although this combination of three different cost functions improves the performance of PSO algorithm in noisy scenarios, it also has restricted the approval of new global and local best particles to some extent and it may slow down the search for optimum vector of parameters even in high SNR cases. So there will be a tradeoff between convergence speed and probability of algorithm failures in noisy environments.

In Practice, one general weakness of PSO is that its local search is not guaranteed convergent and based on the selected swarm size and its initialization; it may converge in a local non-optimal solution. Also, the relatively slower rate of convergence of combined fitness function (13) requires larger swarm sizes and more iterations of algorithm. Hence, some modifications in original position update (11) have introduced using a similar approach suggested in [2]. We modify (11) as follows hybrid update equation:

$$p_i(n) = k_1(p_i(n-1) + \overline{\text{vel}}_i(n)) + k_2(\rho \nabla_{\mathbf{c}_i} J_{r2}). \quad (14)$$

Here, ρ is a small constant about 0.25 and k_1 , k_2 are two constants that may dynamically change during the

algorithm. The added $\nabla_{\mathbf{c}_i} J_{r2}$ term is the gradient of J_{r2} w.r.t multichannel equalizer coefficients. Further details about derivation of this gradient are just the same as the steepest descent method of [1]. In our implementation we have used larger starting k_2 and after few iterations set $k_1 > k_2$. With this choice, we are able to use the gradient as initial fast direction of swarm toward desired point in search space. Later when the swarm finds its way toward desired solution, we set $k_1 > k_2$ in order to let random cooperative search of swarm perform its fine tuning of equalizer coefficients. Clearly the relation $k_1 + k_2 = 1$ must be hold. The exact values of these two will not seriously affect the results, since the PSO will adjust the initial direction of the gradient anyhow. For example a good initial choice would be $k_1 = 0.2, k_2 = 0.8$. These values will be changed into $k_1 = 0.8, k_2 = 0.2$ after 10 iterations.

From now on, we will refer to this modified version of PSO as hybrid PSO.

4. Simulation Results

In this section we compare performance of the proposed hybrid PSO of (14) with the steepest gradient method of [1] for the source extraction step of the well-known inverse filtering approach. Since the comparison with [3] was impossible due to its abstruse presentation of update equations we compare our proposed modified version with standard implementation of PSO (as in (11)) in terms of computational complexity, fidelity, and robustness against noise. We chose J_{42} as objective function of steepest gradient algorithm and used (13) as fitness function of PSO. A 2×2 transfer function $\mathbf{F}(\mathbf{z})$ was chosen as

$$\begin{bmatrix} .65 - .32z^{-1} + .65z^{-2} - .33z^{-3} & .61 + .37z^{-1} - .26z^{-2} + .88z^{-3} \\ .38z^{-1} + .84z^{-2} + .32z^{-3} & .44z^{-1} + .25z^{-6} \end{bmatrix} \quad (15)$$

Inputs $\{x_j(k), j = 1, 2\}$ of MIMO system are mutually independent, zero mean i.i.d., binary sequences taking values ± 1 with probability 0.5 each. The normalized fourth-order cumulant is -2 in this case.

In the first simulation, the equalizer length was chosen to be 15 ($L = 15$). For the purpose of impulse response estimation and extracted signal cancellation, $\hat{f}_{ij0}(\tau)$ was estimated for $-20 < \tau < 20$. The possible permutation and scaling ambiguities in this estimation were solved by imposing proper normalization, and some shifting and alignments as in [1].

The performance index used for evaluation of M_c monte carlo runs is normalized mean square error (NMSE) which can be written for each subchannel $F_{ij}(z)$ as

$$\text{NMSE}_{ij} = \frac{M_c^{-1} \sum_{l=1}^{M_c} \left[\sum_{\tau=-1}^{10} (f_{ij}^l(\tau) - f_{ij}(\tau))^2 \right]}{\sum_{\tau=-1}^{10} (f_{ij}(\tau))^2}. \quad (16)$$

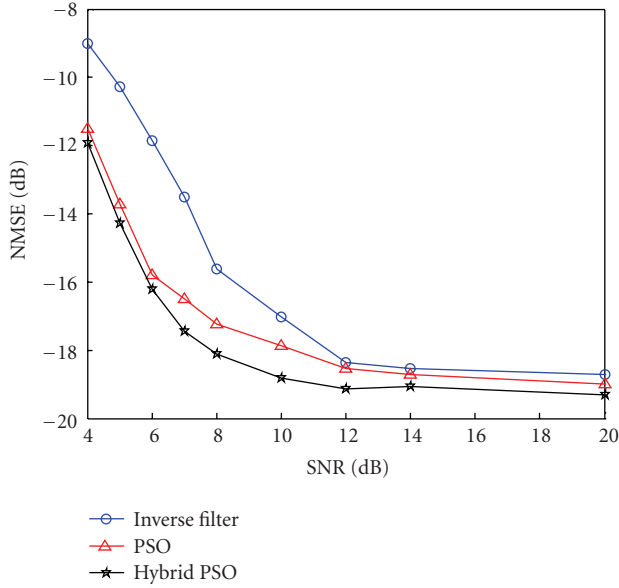


FIGURE 3: Comparison of NMSE for three algorithms in different SNRs.

In which $f_{ij}^l(\tau)$ denotes the estimate of the ij th subchannel in l th Monte Carlo runs. The overall NMSE is:

$$ONMSE = (MN)^{-1} \sum_{i=1}^N \sum_{j=1}^M NMSE_{ij}. \quad (17)$$

The parameters for PSO and hybrid PSO algorithms were chosen according to the tradeoff between convergence speed and algorithm runtime. A population of 100 particles is used with the maximum of 400 allowable iterations for PSO and 60 iterations for hybrid PSO. α and β of (13) are set to 0.3 and acc_1 , acc_2 and ω of (10) are all set to 1. k_1 of (14) was chosen to be 0.2 at the start of simulation ($k_2 = 0.8$) in order to force the swarm in desired direction. After about 30 iterations, k_1 was set to 0.8 ($k_2 = 0.8$) so the cooperative random local search (around the global maximum) of the directed swarm begins.

Figure 3 shows the results for 30 Monte Carlo runs of these three algorithms for several SNRs. The moderate improvement of about 5 dB at lower SNRs can be seen in Figure 3. However as the SNR increases these entire three algorithms exhibit approximately the same very good performance of less than -17 dB.

As it was mentioned previously, the steepest gradient approach has difficulties coping with smaller lengths of available data records. In order to evaluate our proposed algorithm under such conditions, another simulation was done with different number of available observed samples. 30 Monte Carlo simulations were run at SNR = 20 dB for data sample sizes of 200 to 1000 and the results are presented in Figure 4. It can be seen that the significant improvement of about 5 dB has achieved by proposed hybrid PSO algorithm for $300 < T < 600$. Although, in the case of $T = 200$. The standard PSO algorithm is the only method with acceptable performance of less than -10 dB. The poor performance of

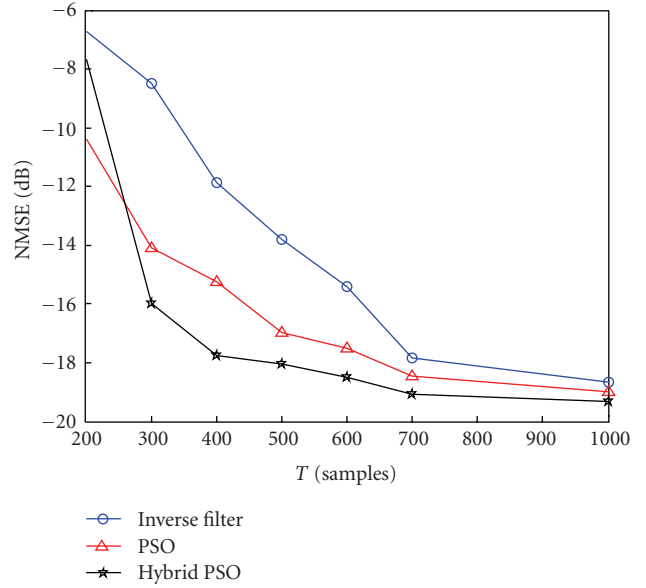


FIGURE 4: Comparison of NMSE of channel impulse response Estimations for different lengths of available data records(SNR = 20, equalizer length = 15).

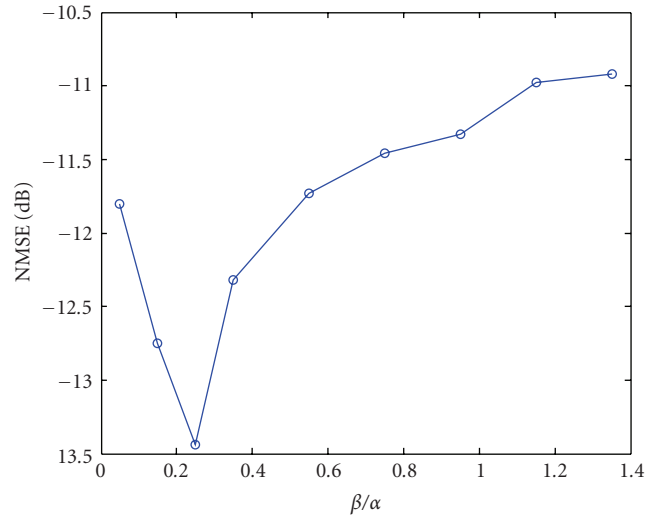


FIGURE 5: The effect of parameters in (13) in identification quality.

hybrid method is due to primary effect of gradient term in (14). Since this gradient has gone in a wrong direction (just like we expect from complete failure of steepest gradient approach), its justification to PSO decreases the probability of even random convergences of PSO search to separation point.

It is interesting to note the stability of PSO method, specially the hybrid PSO. If we define the NMSEs of larger than -10 dB as algorithm failures, even in low noise scenarios, there exists the possibility of failure for steepest gradient approach. For instance, in SNR = 10, although the mean NMSE of 100 Monte Carlo runs are approximately the same, but 3 complete failures have met (10%). Table 1,

TABLE 1: Standard deviation of NMSE of channel impulse response estimations for three methods (SNR = 10, $T = 1000$).

Algorithm	Steepest gradient	Standard PSO	Hybrid PSO
Mean (dB)	-16.8	-17.9	-18.7
Std (dB)	-3.43	-2.1	-0.56
Std/Mean	20.2%	11.9%	2.8%
Best (dB)	-19.41	-19.2	-19.33
Worst (dB)	-8.96	-12.1	-14.23
Number of complete failures	7	5	0

summarizes the standard deviation of NMSE in the case of SNR = 10 and $T = 1000$, for these methods.

In final comparison of PSO and proposed hybrid PSO, the speed of convergence and computational complexity of these two must be studied. Simulation results show that for hybrid PSO, usually a small population of less than 30 particles is enough to very fine convergence of the swarm to the global optimal point; however it takes more than hundreds of iterations for even large populations of more than 100 particles for standard PSO. So, the very fast convergence and the preciseness of optimization are two most important promises of hybrid PSO.

In order to study the effect of α and β of (13) in the identification quality, another experiment was done. In this experiment, the quotient of α and β is the parameter under study. The simulation results for 50 Monte Carlo simulations with SNR = 5 dB are presented in Figure 2. It can be seen that the best results are met by $\beta/\alpha = 0.25$.

Finally, it should be mentioned that our implementation of both PSO and hybrid PSO was the simplest possible approach in order to keep computational complexity and algorithm run time as close as possible to steepest gradient approach. However, it is always possible to further improve PSO algorithms by employing larger population of swarms and more number of allowable iterations with the cost of more computational complexity. Also the performance of any PSO algorithm can be further improved by strategically selecting the starting positions of the particles [8].

5. Conclusion

Two different realizations of Particle Swarm Optimization for source extraction step of well known inverse filtering MIMO identification approach were studied. They both show satisfying results as the original steepest descent method in noise-free scenarios. However, they achieved moderate improvement in lower SNRs and smaller data lengths.

Also the Hybrid PSO algorithm exhibited significant improvement in convergence speed compared to standard PSO, while the initial population of particles was kept smaller. This was the main advantage of hybrid PSO over standard PSO beside the fact that Hybrid PSO was the most precise method with the least probability of complete failure.

Acknowledgment

We gratefully acknowledge partial financial support of this research by the French *Institut national de recherche en informatique et en automatique (INRIA Bordeaux SudOuest)*.

References

- [1] J. K. Tugnait, "Identification and deconvolution of multi-channel linear non-Gaussian processes using higher order statistics and inverse filter criteria," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 658–672, 1997.
- [2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, November 1995.
- [3] T. Song, J. Xu, and K. Zhang, "Blind MIMO channel identification using particle swarm algorithm," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 444–447, Shanghai, China, September 2007.
- [4] D. J. Krusienski and W. K. Jenkins, "A particle swarm optimization-least mean squares algorithm for adaptive filtering," in *Proceedings of the 8th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 241–245, Pacific Grove, Ca, USA, November 2004.
- [5] D. J. Krusienski and W. K. Jenkins, "The application of particle swarm optimization to adaptive IIR phase equalization," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 2, pp. 693–696, Montreal, Canada, May 2004.
- [6] S. Das and A. Abraham, "Synergy of particle swarm optimization with evolutionary algorithms for intelligent search and optimization," in *Proceedings of IEEE International Congress on Evolutionary Computation*, vol. 1, pp. 84–88, 2006.
- [7] J. K. Tugnait, "On blind separation of convolutive mixtures of independent linear signals in unknown additive noise," *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 3109–3111, 1998.
- [8] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 3, pp. 2309–2312, Budapest, Hungary, July 2004.