# A PSO Approach for Non-linear Active Noise Cancellation

HAMIDREZ MODARES ,ALIREZ AHMADYFARD and MOHAMMAD HADADZARIF
Department of Electrical Eng and Robotics
Shahrood university of technology
IRAN

*Abstract*: - In this paper, we address the Active Noise Cancellation (ANC) as a non-linear control problem. The controller of system is designed, using a multi layer perceptron neural network. The neural weights are adapted based on minimization of the measured noise at silence region. We propose a new method based on Particle Swarm Optimization (PSO) to determine the network weights in an adaptive manner.
The modification of PSO algorithm was conducted to the noise cancellation system, in order to handle sudden change of the input noise characteristics. In contrast to the conventional gradient descent type algorithms, the proposed method does not require the estimation of the secondary path parameters. This not only reduces the computational complexity of system, it also improves the stability of ANC system, especially where the secondary path requires a non-linear model. Another advantage of the proposed system is that the adaptation algorithm needs no change when the structure of controller is modified.

*Key-words*: -Active noise cancellation, PSO, Adaptive filtering, Optimization, Neural network, non-linear control

## 1 Introduction

Acoustic noise cancellation is essential from the point of view of health. Long exposures to high level of noise causes serious health hazards to human being.
Active noise control technique (ANC) reduces noise based on the destructive interference of propagating acoustic waves. The basic idea of ANC is to generate a signal (secondary noise), that is equal to a disturbance signal (primary noise) in amplitude and frequency, but has opposite phase. These two signals results in the cancellation of the primary (unwanted) noise in the silence area [1].
Figure 1 shows the block diagram of an adaptive filter which is basis for ANC.
The acoustic noise signal x(n) generated at the source (e.g. an engine or a shaker) propagates in primary path with the transfer function $P_1(z)$ and results noise signal d(n) at silence area. This noise is reduced by interfering signal y(n). The later signal is generated by the appropriate controller output u(n) and sending it through the secondary path with the transfer function $P_2(z)$ .
The remaining difference noise e(n) is measured by a sensor (error microphone), and it is used to change filter coefficients.

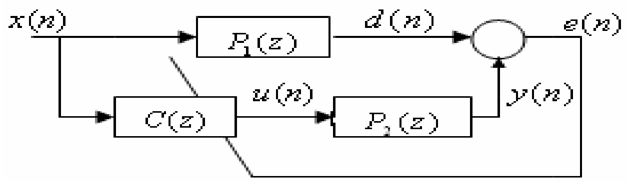


Fig 1: ANC using an adaptive filter

The digital filter C(z) calculates its output by using a reference x(n) and adjustable filter coefficients, or weights. The filter coefficients are updated adaptively, aiming to minimize the mean squared error of noise at silence area.
FIR filters are widely used in ANC systems [1] but linear controllers may not perform well in cases where nonlinearities are found in ANC system. The use of neural networks (NN) for Active control of nonlinear systems has been reported in the literature [2].The ability of an artificial neural network to perform some desired task using the gradient descent based back-propagation algorithm has been reported [3]. It would seem likely, therefore, that such an architecture/algorithm combination could be employed to perform the previously mentioned nonlinear active control tasks, where the neural network would be trained to derive an output signal

which would "cancel" the unwanted noise. As with the linear filter based systems, however, implementation of a gradient descent algorithm in an adaptive feed-forward active control system is not straightforward. Referring to Fig. 1, the reason for complication is dependency of secondary path between the control signal and the associated error measurement. This transfer function incorporates the frequency response characteristics of the control actuator(s) and error sensor(s), as well as the response characteristics of the structural /acoustic system which separates them, including delays due to the finite distance between source(s) and sensor(s). It is intuitively obvious that the existence of this transfer function must be taken into account in adaptive control. This is a fact which is well documented in the literature of both adaptive signal processing [4], and active noise and vibration control [1], and recently restated in regard to neural network based systems. For the (most common) linear FIR filter-based active noise or vibration control arrangement, this leads to a version of the (most common) gradient descent-based least mean square (LMS) algorithm referred to as the filtered-x LMS algorithm [1], [5] , [6]. Stability is maintained in this adaptive algorithm by "filtering" the reference signal, which had been used in deriving the control signal, through an estimate of the secondary path transfer function before it is used by the adaptive algorithm to update the weights in the FIR filter. There are two major problems with using these gradient descent type algorithms for nonlinear filters: first, the robustness of stability for this algorithm is strictly requires an accurate and fast estimation of the secondary path, and second nonlinear filters that use these algorithms are often difficult to implement because of their computational complexity.

In this paper we propose a method based on PSO algorithm for extracting the weights of adaptive NN in ANC system. As the main advantage, the proposed algorithm does not require the estimation of the secondary path, so there is a huge complexity reduction compared with gradient descent algorithms.

This paper will focus on the active noise control problem for non-linear response of an unknown primary acoustic path. The primary path exhibits non-linear distortion when the primary noise propagating in a duct has high sound pressure [7].

## 2 Particle swarm optimization

The particle swarm optimization (PSO) algorithm [8, 9] is an evolutionary computation technique, which is inspired by social behavior of swarms. This approach can be used for a wide range of applications with specific requirement [10].

Similar to the other evolutionary algorithms, PSO is initialized with a population of random solutions. Each potential solution, call *particles*, flies in the D-dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. The location of the *i*th particle and its velocity in tth iteration are denoted by $\vec{X}_i(t)$ and $\vec{V}_i(t)$ respectively. The best previous position (which giving the best fitness value) of the *i*th particle is recorded and represented by symbol $pbest_i$ (personal best) and its location represented as $\vec{x}_{pbest_i}$. The index of the best $pbest_i$ among all the particles is represented by the symbol $gbest$ (global best) and its location presented as $\vec{x}_{gbest}$.

The particle swarm optimization concept consists of, at each time step, changing the velocity and location of each particle toward its $\vec{x}_{pbest_i}$ and $\vec{x}_{gbest}$ locations according to the equations (1) and (2), respectively

$$\vec{v}_i(t) = \varphi\, v_i(t-1) + r_1 c_1 (\vec{x}_{pbest_i} - \vec{x}_i(t)) + r_2 c_2 (\vec{x}_{gbest} - \vec{x}_i(t)) \tag{1}$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \tag{2}$$

Where $\varphi$ is inertia weight, $c_1$ and $c_2$ are acceleration constants, $r_1$ and $r_2$ are random parameter in the range [0, 1]. The first term in equation (2) represents the inertia of velocity at previous iteration; the second part is the "cognition" part, which represents the private thinking by itself; the third part is the "social" part, which represents the cooperation among the particles [11]. The steps of PSO algorithm is as follows:

1- Set iteration counter t=1. Initialize a population including "p" particles, the ith particle has random location $\vec{X}_i(t)$ and random velocity $\vec{V}_i(t)$ in M dimensional (where M is the number of filter coefficients) space

2- Evaluate the fitness for each particle

3- Compare the evaluated fitness of each particle with its $pbest$ . If current fitness is better than $pbest$ , then set the fitness value as the $pbest$ and its location as the $\vec{x}_{pbest_i}$ . Furthermore, if the particle fitness is better than $gbest$ , then set the fitness value as $gbest$ and its location as the $\vec{x}_{gbest}$

4- Change the velocity and location of the particle according to the equations (2) and (3), respectively

5- t=t +1, go to step b until a stop criterion is met, Usually a sufficiently good fitness value or exceeding the number of iterations above predefined maximum tmax, are used as the stop criterion

## 3 Adaptive PSO algorithm for ANC

The block diagram and the operation steps are shown in Fig. 2 and 3 respectively. First we choose "P" particles in an M-dimensional space (M denotes the number of weights of NN) as random points (step 1 in Fig 3). The location of each particle determines a candidate MLP NN. The reference noise signal, x(n), is filtered with a particle (MLP NN) at a time based on the header equation in Fig.3 calculate the output of MLP NN. The particles are selected rotationally after every W samples of reference signal (Fig. 2). The fitness of each particle ($F(s_i)$ in Fig.2) is measured from W samples of error signal e(k), obtained by the microphone, in RMS sense (step 2 in Fig.3). Once all the filters corresponding to the particles in population are used, algorithm compares the evaluated fitness value of each particle with its personal best ( $pbest$ ). If current fitness value is better than $pbest$ , then it set the current value as the $pbest$ and the current location as $\vec{x}_{pbest}$. Furthermore, if current is better than $gbest$ , then set the current value as the $gbest$ and the current location as the $\vec{x}_{gbest}$ (steps 3 and 4 in Fig.3).

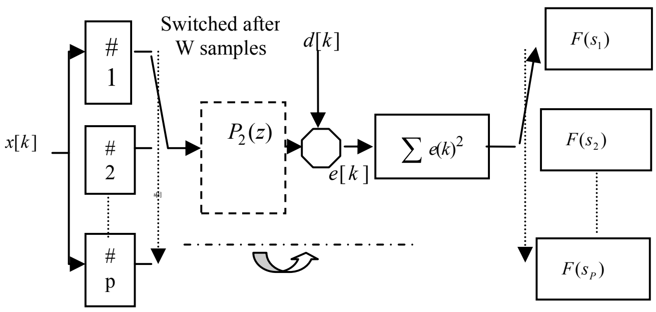


Fig.2 Block diagram of the single channel ANC using PSO algorithm

**PSO Adaptive NN filter for ANC system**

1. generation of particles as random points (p is number of particles in search space) at iteration t=0:

$$\vec{x}_i(t) = [C_i(0), C_i(1), ...., C_i(M-1)] \qquad 1 \le i \le p$$

(M is number of neural network weights)

2. fitness evaluation of each particle:

$$F(s_i) = 1 / \sum_{n=0}^{W-1} e_i^{\,2}(n)$$

3. Compare the evaluated fitness of each particle with its $pbest_i$:

If F($s_i$)> $pbest_i$

(a) $pbest_i$=F($s_i$)

(b) $\vec{x}_{pbest_i} = x_i(t)$

4 . Compare the evaluated fitness of each particle with $gbest$ :

If F($s_i$)> $gbest$

(a) $gbest$ = F ($s_i$)

(b) $\vec{x}_{gbest} = x_i(t)$

5. Change the velocity and location of the particle according to the equations (1) and (2)

6. go to step 2 and continue

Fig.3 operation steps of the adaptive PSO algorithms for ANC system

After above comparison, PSO algorithm Change the velocity and location of the particle according to the equations (2) and (3) (step 5 in fig.3) and then go to step 2 of algorithm for generating another iteration of algorithm

The result of experiments using the original PSO algorithm showed that the algorithm does not respond properly to sudden changes in noise source characteristics. It seems the algorithm traps in non-optimal points based on global and personal memories ($gbest$ and $pbest$ ). The result showed, particles bounce around the $gbest$ with low velocities. If a sudden change in input noise parameters (frequency or power) occurs, the algorithm failed to find global optimum.

We modified the standard algorithm to overcome the addressed problem. As we will see, the modified PSO

is very successful in finding optimal solutions, even when the change in input noise is significant. The modification of PSO algorithm is as follow.
First we used $gbest$ in the search space, as a sentry, to test for changes in the noise source. For this purpose, we introduce a new coefficient, $\Delta$ as follow:

$$\begin{aligned} & if\ F(gbest(t)) \geq \alpha\, F(gbest(t-1)) \Rightarrow \\ & \Delta = 0 \\ & else\ if\ F(gbest(t)) < \alpha\, F(gbest(t-1)) \Rightarrow \\ & \Delta = \frac{F(gbest(t-1)) - F(gbest(t))}{F(gbest(t-1))} \end{aligned} \quad (3)$$

The above strategy shows that if the fitness of $gbest$ (sentry particle) at the current iteration is smaller than this parameter in the previous iteration (e.g. the input noise properties has been changed), $\Delta$ will have a value between 0 and 1, depending on the input noise change. So by inspecting $\Delta$ on each iteration of algorithm, the significant changes in input noise are detected. The parameter $\alpha$ controls the process in a way that if change in input noise is moderate, the standard PSO being used.
After detecting a significant change in the input noise, the particles in population must forget their own global memories to find the new global best ($gbest$).
To follow this idea, we decrease the fitness of particles at the rate of a big evaporation constant, once $\Delta$ shows a significant change. As a direct result of this, other particles have chance to have the fitness bigger than the previous $gbest$ and $pbest$ .
We also increase the velocity of particles after detecting significant changes in the input noise to let them search a bigger solution space for optimal solution. We formulated this idea as follow and perform it, on each iteration of algorithm:

$$\begin{aligned} & if\ \ \Delta > 0\ \ then \\ & (a)\ \begin{cases} gbest = gbest \times T \\ pbesti = pbesti \times T \end{cases} \\ & (b)\ \varphi = \phi + \eta\,\Delta \end{aligned} \quad (4)$$

The parameter T is evaporation constant selected between 0 and 1 and $\eta$ is a constant set to bigger than 1. Equation (4) shows that if the noise source changes ($\Delta$ >0), the fitness of $gbest$ and $pbest$ evaporated at the rate of the T and also the velocity of particles increase by $\eta\Delta$ only in one iteration.

# 4 The experimental results

We designed an ANC system with a non-linear model in which a MLP neural network used as the adaptive controller.
The neural network weights are updated using the proposed PSO algorithm.
We simulated the noise cancellation process in the above system and compared it with the results when the filtered-x back-propagation (FX-BP) determines the neural network weights.
In this simulation we modeled the input noise as mixture of two components: a 200Hz sinusoidal signal and a Gaussian white noise. This synthetic signal can model properly the acoustic noise in industrial environments.
Figs 4(a) and 5(a) show the process of noise cancellation at the silence area for FX-BP and the proposed method respectively.
The results are represented in frequency domain in Figs 4(b) and 5(b). Before analyzing this results, we need to explain that although the input noise has a main component at 200Hz, when it pass through the non-linear primary path, the second harmonic of signal at 400Hz is also generated.
Figs 4(b) and 5(b) show that the proposed method better cancel the noise component at 400Hz, than FX-BP method.
The noise cancellation difference at this frequency is about 10dB.
The results of cancellation at primary component 200Hz, is quiet similar for two methods.
A major superiority of the proposed algorithm to FX-BP is robustness of the algorithm when the input noise characteristics change significantly. As shown in Fig 6 where the input noise power was increased 12.5 times, FX-BP algorithm fails to converge after the input noise power increased (Fig 6(a)) where the proposed method successfully converge to the new optimum solution (Fig6 (b)).

# 5 Computational complexity of the proposed algorithm

For the proposed algorithm, the number of addition and multiplication for each updating of every nonlinear filter coefficients (weights) are represented in Table 1. L is the number of filter coefficients (weights) and P is the number of particles in PSO population. This is very low computational complexity in compared with the gradient descent algorithms. In order to prove our claim, the volterra and bilinear filters coefficients (weights) that is presented in [12] are compared in Table 2. Our method owes it simplicity to the fact that we don't change the filter coefficients (weights) for each input samples; instead we update the coefficients The low computational complexity of our algorithm is a major advantage for it. (weights) after evaluation of all particles in population which takes $W \times P$ samples.

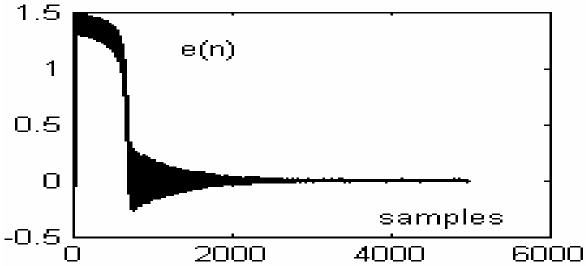


Fig. 4(a) Error signal for ANC system using FX BP algorithm

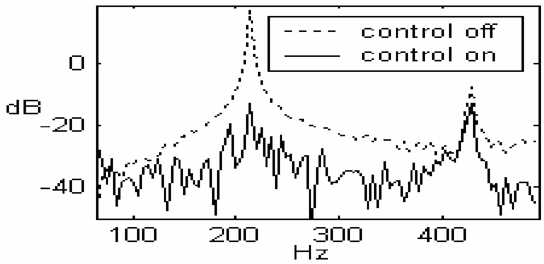


Fig. 4(b) Power spectrum of active noise canceling errors using FX BP algorithm

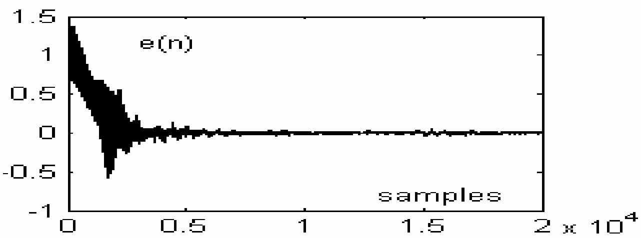


Fig. 5(a) Error signal for ANC system using PSO Algorithm

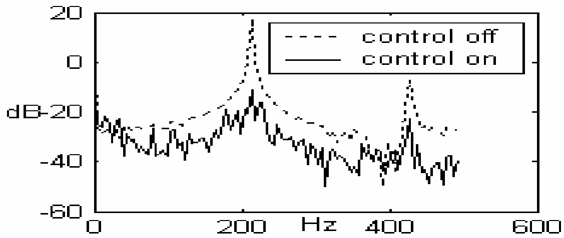


Fig. 5(b) Power spectrum of active noise canceling errors using PSO algorithm

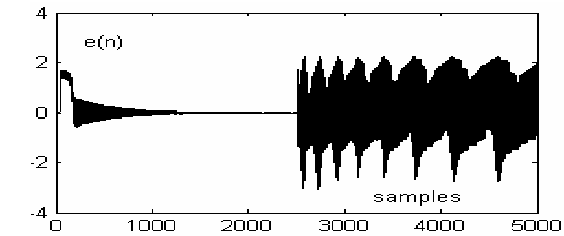


Fig. 6(a) Error signal for ANC system with sudden change in power, using FX BP algorithm

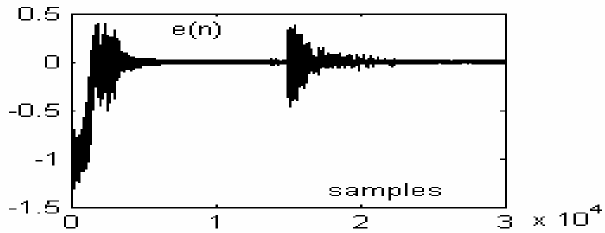


Fig. 6(b) Error signal for ANC system with sudden change in power, using PSO algorithm

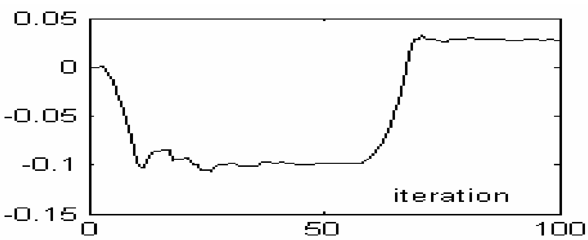


Fig. 6(d) one of filter coefficients , over PSO iterations for noise with a sudden change in frequency

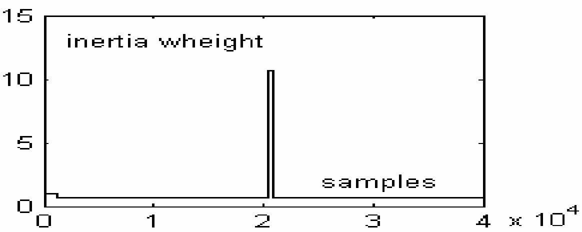


Fig. 6(c) inertia weight of PSO algorithm for noise with a sudden change in frequency

| calculations | Number of additions | Number of multiplications | Per Number of input samples |
|---|---|---|---|
| equation 1 | 4LP | 2LP | WP |
| equation 2 | LP | 0 | WP |
| Total | 5LP | 2LP | WP |

Table 1. Computational complexity of PSO algorithm for updating each nonlinear filter coefficients

| Algorithm/filter | Number of addition | Number of multiplication | Per number of input samples |
|---|---|---|---|
| PSO/volterra | 1600 | 640 | 250 |
| FX-LMS/ Volterra | 2080 | 4225 | 1 |
| PSO/bilinear | 1600 | 640 | 250 |
| FX-LMS/ blinear | 4289 | 4356 | 1 |

Table 2. Comparing computational complexity of PSO with FX-LMS algorithm for volterra and bilinear filters (L=64, P=5, W=50)

This makes the real time implementation possible.

## 6 Conclusions

A new adaptive PSO algorithm has been presented for active noise control applications. The main advantage of this type of algorithm is that it does not require estimation of the secondary path estimation. Proposed algorithm has smaller computational complexity compared with gradient descent algorithms. Besides, the convergence speed depends on the number of particles, so small population produce fast convergence but even less than the convergence speed of the FX-BP algorithm.

An advantage associated with the PSO is that various filter structures can be used with no change to the adaptation algorithm, which allows quick selection of appropriate filter structure for the problem in hand.

*References:*


[1] S.J.Elliot and P.A. Nelson , Active Noise Control . *IEEE signal processing magazine*, 1993

[2] S. D. Snyder and N. Tanaka, "A neural network for feedforward controlled smart structures*," J. Intell. Mat. Sysf. Struct.,* Vol. 4, pp. 373-378, 1993.

[3] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. I : Foundations. Cambridge, MA MIT Press, 1986.

[4] D. R. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path*," IEEE Trans. Acoustic. Speech Signal Process*, vol. ASSP-28, pp. 454-467, 1980

[5] B. Widrow and S. D. Steams*, Adaptive Signal Processing. Englewood Cliffs*, NJ: Prentice-Hall, 1985

[6] S. J. Elliott and P. A. Nelson, "The application of adaptive filtering to the active control of sound and vibration*," ISVR Tech. Rep*., vol. 136, 1985

[7] L. Tan, J. Jiang, Adaptive volterra filters for active control of nonlinear noise processes*, IEEE Transactions on Signal Processing* 49 (2001) 1667– 1676.

[8] M. Clerc, J. Kennedy, The Particle Swarm, Explosion, Stability, and Convergence in Multi-Dimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, Vol. 6, pp 58-73, 2002.

[9] J. Kennedy, R. Eberhart., Particle swarm optimization. Proc. *IEEE Int. Conf. on Neural Networks*, 1995: 1942-1948

[10] R. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources. *IEEE Int. Conf. on Evolutionary Computation*, 2001: 81-86

[11] J. Kennedy, The particle swarm: social adaptation of knowledge. *IEEE Int. Conf. on Evolutionary Computation*, 1997: 303-308

[12] Sen M. Kuo, Hsien-Tsai Wu, Nonlinear Adaptive Bilinear Filters for Active Noise Control Systems. *IEEE Transactions on circuits and systems*, Vol. 52, No.3, 2005