Efficient Scheme of Pole-Zero System Identification using Particle Swarm Optimization Technique

Babita Majhi, G. Panda, *Senior Member IEEE* and A. Choubey

***Abstract*— This paper introduces the application of Particle Swarm Optimization (PSO) technique to identify the parameters of pole-zero plants or infinite impulse response (IIR) systems. The PSO is one of the evolutionary computing tools that performs a structured randomized search of an unknown parameter space by manipulating a population of parameter estimates to converge to a suitable solution with low computational complexity. This paper applies this powerful PSO tool to identify the parameters of standard IIR systems and compares the results with those obtained using the Genetic Algorithm (GA). The comparative results reveal that the PSO shows faster convergence, involves low complexity, yields minimum MSE level and exhibits superior identification performance in comparison to its GA counterpart**

I. Introduction

In recent years, many research activities are underway to extend the finite impulse response (FIR) filter into the more general infinite impulse response (IIR) configuration that offers potential performance improvements and less computational cost than the equivalent FIR filters[1]. As the error surface of an adaptive IIR filters is multimodal with respect to the filter coefficients, the learning algorithms for IIR filters can easily be stuck at local minima and can not converge to the global optimum[2]. A few modifications to gradient decent algorithms exists that can improve the performance such as adding noise to the gradient calculation or using the equation error[11] adaptation. Identification of pole-zero or IIR plants finds extensive applications in control, instrumentation, acoustic noise and vibration control and telecommunication. Multi layer perceptron (MLP) has been used for identification of pole-zero systems[12]. However the MLP is associated with some inherent draw backs, e.g., multiple local minima problem, the difficulty of selecting the number of hidden units and the possibility of over fitting.

In recent past the GA has been employed for multimodal optimization in adaptive IIR filtering, system identification and control[3]-[7]. However, the disadvantages of using GA are slow convergence and high computational complexity. Again there are only two key operations i. e. crossover and mutation involved in GA to overcome the trap situation, but there is still some possibility that the solution is trapped in local minima.

The PSO on the other hand is another structured stochastic search algorithm that has recently gained popularity for multimodal optimization problems and adaptive IIR filtering[8]-[10]. The PSO does not use evolution operators such as crossover and mutation although it shares many similarities with the GA. The PSO emulates the swarm behaviour of insects, animal herding, birds flocking and fish schooling where these swarms search for food in a collaborative manner. Each member in the swarm adapts its search patterns by learning from its own experience and other members' experiences. Each member in the swarm, called a particle, represents a potential solution in the search space. The global optimum is regarded as the location of food. Each particle has a fitness value and a velocity to adjust its flying direction according to the best experiences of the swarm to search for the global optimum.

In this paper we have chosen the PSO as a candidate for adapting coefficients of the pole-zero systems because it offers faster convergence during training and it is computationally involves low complexity as compared to GA. Performance of the proposed PSO based model is compared with its GA counterpart by simulating standard IIR systems. Simulation results exhibit that the PSO based training of parameters offers better and faster pole-zero system identification. Rest of the paper is organized as follows :

Section II discusses the pole-zero identification problem. Basics of particle swarm optimization is dealts in Section III. In Section IV the PSO based update algorithm is developed for IIR system identification. For performance evaluation, the simulation study is carried and the results are presented in Section V. Finally, conclusion of the paper is outlined in Section VI.

II. System Identification with IIR Direct-Form Realization

In the system identification configuration, the adaptive algorithm identifies the pole-zero coefficients of the adaptive filter such that its input/output relationship matches as close as that of the unknown system.

B. Majhi and G. Panda are with Department of Electronics & Communication Engineering, National Institute of Technology, Rourkela-769008, Orissa, INDIA (e-mail: babita.majhi@gmail.com, ganaapati.panda@gmail.com).

A. Choubey is with the Department of Electronics Engineering, National Institute of Technology, Jamshedpur – 831014, INDIA (e-mail: arvindnit@gmail.com).

978-1-4244-1823-7/08/$25.00©2008 IEEE

Fig. 1 shows the general block diagram of an adaptive system identifier where the unknown system or plant is descried by

$$y_o(n) = \left[\frac{B(q^{-1})}{A(q^{-1})}\right] x(n) \quad (1)$$

$$y(n) = y_o(n) + v(n) \quad (2)$$

where $A(q^{-1}) = 1 - \sum_{i=1}^{n_a} a_i q^{-i}$ and $B(q^{-1}) = \sum_{j=0}^{n_b} b_j q^{-j}$ are

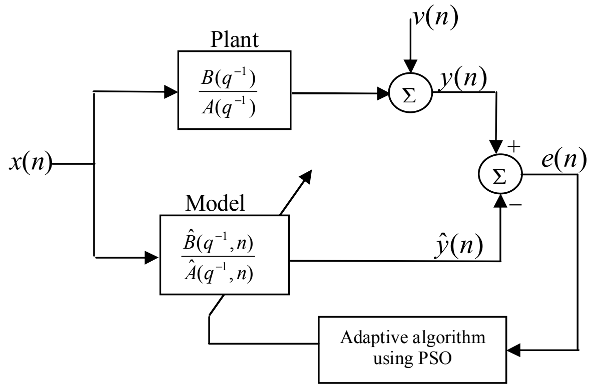


Fig. 1 Block diagram of an adaptive pole-zero system identifier

co-prime polynomials of the unit delay operator q^{-1}, and $x(n)$ and $v(n)$ are the input signal and the additive Guassian noise respectively. The adaptive filter is implemented with the direct form structure described by

$$\hat{y}(n) = \left[\frac{\hat{B}(q^{-1}, n)}{\hat{A}(q^{-1}, n)}\right] x(n) \quad (3)$$

where,

$$\hat{A}(q^{-1}, n) = 1 - \sum_{i=1}^{\hat{n}_a} \hat{a}_i(n) q^{-i} \text{ and } \hat{B}(q^{-1}, n) = \sum_{j=0}^{\hat{n}_b} \hat{b}_j(n) q^{-j}$$

The error is given by $e(n) = y(n) - \hat{y}(n)$ (4)

The PSO is employed in this paper to estimate A and B such that the responses $y(n)$ and $\hat{y}(n)$ match as closely as possible.

III. Basics of Particle Swarm Optimization

Assume in the D-dimensional search space, $i = 1,2,......., N$, the i th particle can be described by a vector as : $X_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$, the velocity of the particle can be expressed as : $V_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$, the best previous position that the i th particle arrived is : $P_i = (p_{i1}, p_{i2}, ..., p_{iD})^T$, The best previous position (the position giving the best fitness value) of the i th particle is recorded and represented as : $P_g = (p_{g1}, p_{g2}, ..., p_{gD})^T$, the biggest velocity is : $V_{\max} = (v_{\max 1}, v_{\max 2}, ..., v_{\max D})^T$, the dynamic range of the particle $X_{\max} = (x_{\max 1}, x_{\max 2}, ..., x_{\max D})^T$, then the basic algorithm of PSO can be described as follows :

1) Initialize particle swarm by adopting the position and the velocity which are created randomly in the whole search space, and make the particle move in the whole problem space in a random way.

2) Compute each particle's fitness value in its current position x_i.

3) Each particle can remember the best position that itself once has arrived.

4) Each particle can perceive the best position that the neighbouring particle has arrived already.

5) While changing the velocity, it considers the best position that itself has once arrived and that the neighbouring particle has already arrived. The velocity and the position of each particle is updated using 5, 6 and 7.

$$v_{id} = w * v_{id} + C_1 * rand_1 * (p_{id} - x_{id}) + C_2 * rand_2 * (p_{gd} - x_{id}) \quad (5)$$

$$v_{id} = \begin{cases} v_{\max d}, & v_{id} > v_{\max d} \\ -v_{\max d}, & v_{id} < -v_{\max d} \end{cases} \quad (6)$$

$$x_{id} = x_{id} + v_{id} \quad (7)$$

where, $d = 1,2,......, D$; C_1 and C_2 are the factor of acceleration, $rand_1$ and $rand_2$ are random numbers between 0 to 1 and the value of inertial weight, w is to be chosen carefully.

6) If it satisfies the predefined termination condition or attains the maximum number of iterations, the computation is terminated, otherwise the algorithm goes to step -2.

Various studies have shown that under certain conditions the convergence of the PSO algorithm is guaranteed to a stable equilibrium point. These conditions include

$$w > \frac{1}{2}(C_1 + C_2) - 1 \textit{ and } 0 < w < 1 \quad (8)$$

IV. PSO Based Pole-Zero Sysytem Identification

The objective of an adaptive algorithm is to change the filter weights iteratively so that the squared error, $e^2(k)$ is minimized and subsequently reduced to a minimum. The updating of the weights of the PSO based model is carried out using the training rule as outlined in the following steps:

1. K (K≥500) number of input signal samples between -0.5 to +0.5 is generated.
2. Each of the input samples is passed through the plant and then contaminated with additive noise of known strength. The resultant signal is the desired signal. In

this way K numbers of desired samples are produced by feeding all the K input samples.

3. The same K input samples is fed simultaneously to the model and K numbers of estimated outputs are obtained.
4. Each of the desired output is compared with the corresponding model output and K errors are produced.
5. The mean square error (MSE) for a given pole-zero parameters (corresponding to n^{th} particle) is determined by using the relation.

$$MSE(n) = \frac{\sum_{i=1}^{k} e_i^2}{k}$$ This is repeated for M times.

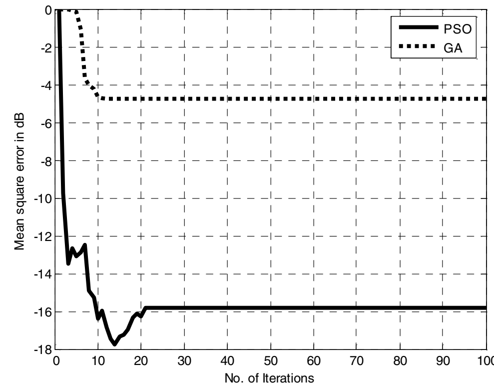
6. Since the objective is to minimize MSE (n), n = 1 to M the PSO based optimization method is used.
7. The velocity and position of each bird is updated using (5) and (7).
8. In each iteration the minimum MSE, MMSE (expressed in dB) is stored which shows the learning behavior of adaptive model from iteration to iteration.
9. When the MMSE has reached the pre-specified level the optimization is stopped.
10. At this step all the particles attains almost the same positions, which represent the desired coefficients of the given pole-zero system.

V. SIMULATION STUDY AND DISCUSSIONS

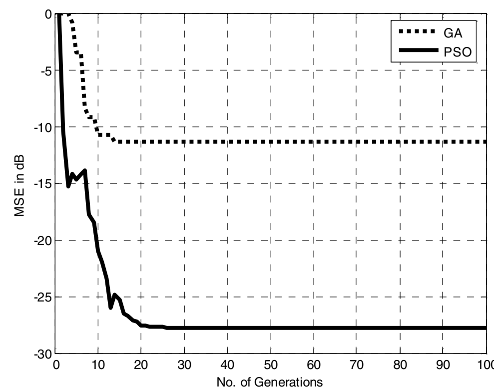
The system identification configuration as shown in Fig. 1 is used for simulation. The plant is considered as one of the benchmark pole-zero system, while the adaptive system is an IIR system of the same order whose coefficients are updated by using GA and PSO based training. White Guassian noise with zero mean and unity variance are added at the output of the plant.

Example -1 : The plant is a second order pole-zero system taken from [4].

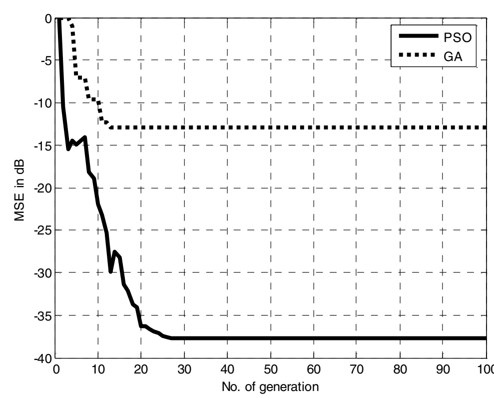
$Plant = \dfrac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}$. The model is given by $\dfrac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1} - a_2 z^{-2}}$. For the GA the parameters used in the simulation study are no. of input samples = 1000, population=80, no. of bits per chromosome=20, probability of crossover = 0.8, probability of mutation =0.01 and $w_{\max} = 1, w_{\min} = -1$. For PSO we use no. of input samples = 500, population = 520, $w = 0.729$, $c_1 = 1.49445$ and $c_2 = 1.49445$. The convergence plot for this example for 10, 20 and 30dB SNRs are shown in Figs. 2(a)-2(c). From these figures it is evident that the PSO provides faster convergence and achieves better noise floor level in comparison to the GA counterpart. For 10dB SNR the PSO based training yields -16dB whereas the GA settles at -5dB. Similarly for 20dB and 30dB SNRs the PSO based technique settles at -28dB and -38dB respectively whereas the GA based method settles only at -12dB and -14dB.



(a)10dB SNR



(b) 20dB SNR

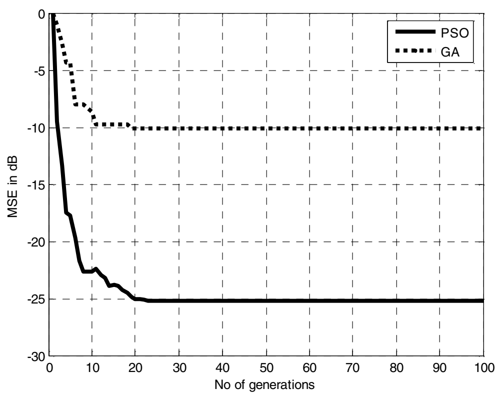


(c) 30dB SNR

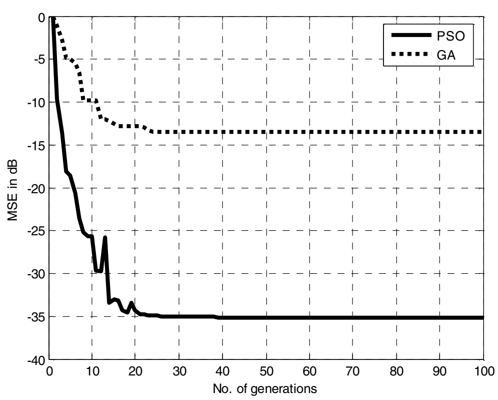
Fig. 2 Comparison of convergence characteristics for Example-1 between PSO and GA with different SNR conditions

Example-2 : The third order pole–zero system used in this example is taken from [4]. The transfer function of the plant is given by $\dfrac{1.2 - 0.5z^{-1} + 0.1z^{-2}}{1 + 0.3z^{-1} - 0.4z^{-2} + 0.5z^{-3}}$ and the model used is

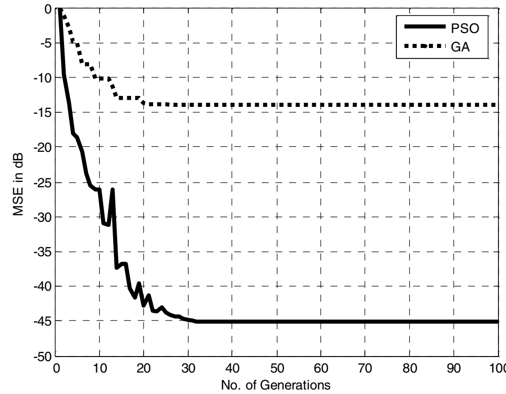
$\frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3}}$. The parameters used for the GA based approach are no. of input samples = 1000, population=80, no. of bits per chromosome=20, probability of crossover = 0.9, probability of mutation =0.01 and $w_{\max} = 1.3, w_{\min} = -1.3$. For PSO we use no. of input samples = 500, population = 240, $w = 0.729$, $c_1 = 1.55$ and $c_2 = 1.55$. Figs. 3(a)-3(c) show the comparison of convergence plots for 10dB, 20dB and 30dB SNRs respectively. From these plots it is observed that for 10dB, 20dB and 30dB SNRs the PSO based training settles at -25dB, -35dB and -45dB whereas the GA based approach settles at -10dB, -12dB and -14dB respectively.



(a) 10dB SNR



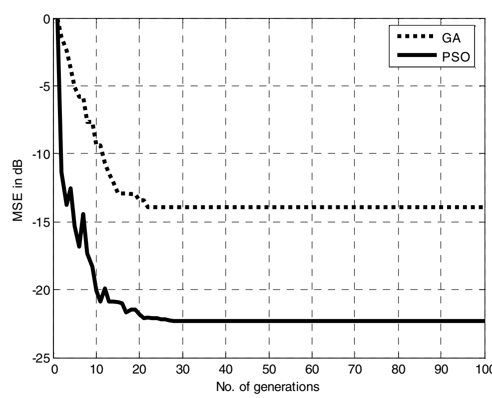
(b) 20dB SNR



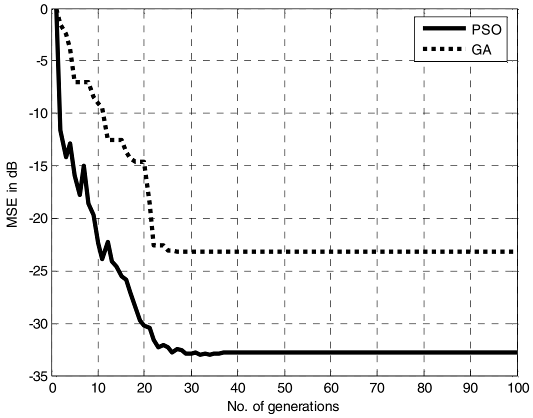
(c) 30dB SNR

Fig. 3 Comparison of convergence characteristics of Example-2 between PSO and GA with different SNR conditions

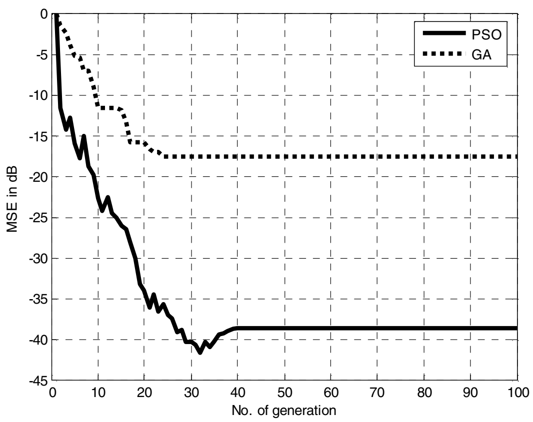
Example-3 : The transfer function of a fourth order system used is taken from [12] and is given by $\frac{(1-0.9z^{-1})(1+0.81z^{-2})}{(1-0.71z^{-1}+0.25z^{-2})(1+0.75z^{-1}+0.56z^{-2})}$. The model is also of the same order and is given as $\frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3} - a_4 z^{-4}}$.For the GA based approach the parameters used are no. of input samples = 1000, population=120, no. of bits per chromosome=60, probability of crossover = 0.9, probability of mutation =0.01 and $w_{\max} = 1.3, w_{\min} = -1.3$. In PSO various values used are number of input samples = 500, population =500, $w = 0.729$, $c_1 = 2$ and $c_2 = 2$. Figs. 4(a)-4(c) show the convergence characteristics plots for 10dB, 20dB and 30dB SNRs respectively. The PSO based training settles at -22dB, -32dB and -38dB for given 10dB, 20dB and 30dB SNRs, whereas the GA based training settles at -13dB, -22dB and -18dB respectively.



(a) 10dB SNR



(b) 20dB SNR



(c) 30dB SNR

Fig. 4 Comparison of convergence characteristics of Example-3 between PSO and GA with different SNR conditions

Table -1
Comparison of estimated parameters of the systems using GA and PSO with 30dB SNR

Order of the system	Actual parameters	Estimated parameters using GA	Estimated parameters using PSO
2^{nd}	0.05	0.0530	0.0505
	-0.4	-0.4894	-0.4011
	1.1314	0.9158	1.1295
	-0.25	-0.0653	-0.2480
3^{rd}	1.2	1.2355	1.2008
	-0.5	-0.4225	-0.4992
	0.1	0.5674	0.1050
	-0.3	-0.3277	-0.2998
	0.4	0.1216	0.3966
	0.5	0.3189	0.4986
4^{th}	1	1.0018	1.0014
	-0.9	-0.6291	-0.8993
	0.81	0.7273	0.8097
	-0.729	-0.5838	-0.7273
	0.23	0.0125	0.2294
	-0.433	-0.5305	-0.4326
	0.339	0.2635	0.3382
	-0.5184	-0.4237	-0.5184

Table -2
Comparison of CPU time and minimum MSE obtained using GA and PSO

Type of plant	CPU Time in second		Minimum MSE obtained	
	GA	PSO	GA	PSO
2^{nd} order	3.4297	2.2358	0.0011	7.5474×10^{-5}
3^{rd} order	5.5050	2.0404	0.0062	7.2921×10^{-5}
4^{th} order	9.1883	4.0771	0.0050	7.3337×10^{-5}

Table-3
Comparison of no. of squared error evaluation between PSO and GA

Type of plant	Nos. of squared error evaluation		saving in evaluation
	PSO	GA	
2^{nd} order	26000000	27446000	144, 6000
3^{rd} order	12000000	26206000	142, 06000
4^{th} order	25000000	38798000	137, 98000

It is observed that in all three examples the PSO based training outperforms the GA both in terms of rate of convergence and residual mean squared error.

The estimated parameters using PSO and GA algorithms for all the three pole-zero systems are given in the Table -1. The values obtained by the PSO based approach are more accurate in comparison to those obtained using GA. The minimum MSE obtained and CPU time shown in Table-2 are observed to be lower in case of PSO based technique than the GA counterpart. Table-3 shows the number of times the error square to be evaluated to obtain the minimum MSE using PSO and GA algorithms. It is evident from this table that in case of PSO based approach the numbers of evaluation is also less in all cases in comparison to its GA counterpart. From all these observations it is clear that the PSO based training is much faster and more accurately estimates the pole-zero coefficients in comparison to its GA counterpart.

VI. CONCLUSION

In this paper we have demonstrated how the PSO can be used to train the pole-zero parameters of various IIR plants. Computer simulation results reveal that the proposed approach is robust and offers faster training compared to that obtained by the conventional GA based method. Further the proposed method is promising and efficient in the sense that it requires less computation and estimates the IIR coefficients more accurately than its GA based counterpart.

REFERENCES

[1] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Prentice-Hall, 1985.
[2] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[3] R. Nambiar and P. Mars, “Genetic and annealing approaches to adaptive signal filtering”, Proc. Of 26th Asilomar conf. on Signals, systems and Computers, vol 2, pp. 871-875, Oct. 1992.
[4] S. C. NG, S. H. leung, C. Y. Chung, a. Luk and W. H. Lau, :The genetic Search approach”, IEEE signal Processing Magazine, pp. 28-46, November 1996.
[5] K. S. tang, K. F. Man and Q. he, “Genetic Algorithms and their applications”, IEEE signal Processing Magazine, pp. 22-37, Nov. 1996.
[6] L. Yao and W. A. Sethares, “Nonlinear parameter estimation via the Genetic Algorithms”, IEEE Trans. On Signal Processing, vol. 42, April 1994.
[7] Kristinn Kristinsson and Guy A. Dumont, “System identification and control using Genetic Algorithms“, IEEE Trans. On Systems, Man and Cybernetics, vol. 22, no. 5, September 1992.
[8] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory”, in Proc.. 6th Int. Symp. Micro-machine Human Sci., Nagoya, Japan, 1995, pp. 39-43.
[9] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, in Proc. IEEE Int. Conf. on Neural Networks, 1995, pp. 1942-1948.
[10] D. J. Krusienski and W. K. Jenkins, ‘Particle swarm optimization for adaptive IIR filter structures”, IEEE congress on evolutionary computation (CEC 2004), vol. 1, 19-23 June, 2004, pp. 965-970.
[11] J. J. Shynk, “Adaptive IIR Filtering”, IEEE ASSP Magazine, pp. 4-21, April 1989.
[12] B. B. Murthy, C. F. N. Cowan and G. Panda, “Efficient scheme of pole-zero system identification on multilayer neural network”, Electronics Letters, vol. 29, issue 1, pp. 73, Jan., 1993.