International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467

Comparing Autoregressive Moving Average (ARMA) coefficients determination using Artificial Neural Networks with other techniques

Abiodun M. Aibinu, Momoh J. E. Salami, Amir A. Shafie and Athaur Rahman Najeeb

***Abstract*—**Autoregressive Moving average (ARMA) is a parametric based method of signal representation. It is suitable for problems in which the signal can be modeled by explicit known source functions with a few adjustable parameters. Various methods have been suggested for the coefficients determination among which are Prony, Pade, Autocorrelation, Covariance and most recently, the use of Artificial Neural Network technique.

In this paper, the method of using Artificial Neural network (ANN) technique is compared with some known and widely acceptable techniques. The comparisons is entirely based on the value of the coefficients obtained. Result obtained shows that the use of ANN also gives accurate in computing the coefficients of an ARMA system.

***Keywords*—**Autoregressive Moving Average, Coefficients, Back Propagation, Model Parameters, Neural Network, Weight.

I. INTRODUCTION

The use of modeling technique to predict or reconstruct a data sequence is concerned with the representation of data in an efficient technique [1]–[4], [6], [10], [13]. Signal modeling have been used in radar application, geophysical application, Medical signal processing, ultrasonic tissue backscatter coefficient estimation, speech processing, music understanding and more recently in the field of Magnetic Resonance Imaging (MRI) reconstruction [2], [4], [6], [10], [13], [14], [16], [17].

Signal modeling involves two steps steps [2], these are;

1) **Model selection**: Choosing an appropriate parametric form for the model data
2) **Model Parameter determination**: This include model order and model coefficients determination.

Despite the success reported in the use of modeling technique, two important problems constitutes challenges to the applicability of this method, these are:

1) **Estimation of Model order**: There have been various effort in determining a workable criteria for the determination of an appropriate model order. The use of a model with an order too high over fits the data while the use of a model with a low order leads to insensitivity to noise [2], [4], [6], [15].
2) **Estimation of model coefficient** : The second important challenges mitigating against the use of modeling technique is the estimation of the model coefficients. Some of the existing methods of determining model coefficients includes Prony, Pade, Least Square, Shank, Autocorrelation, Autocovariance methods [9].

Athaur Rahman Najeeb is with the Kulliyah of Engineering, International Islamic University Malaysia (IIUM), email: athaur@iiu.edu.my

Momoh.J.E Salami is with the Kulliyah of Engineering, International Islamic University Malaysia (IIUM), email: momoh@iiu.edu.my

Amir A. Shafie is with the Kulliyah of Engineering, International Islamic University Malaysia (IIUM), email: aashafie@iiu.edu.my

Abiodun. M. Aibinu, Malaysia, email: maibinu@gmail.com

The organization of this paper is as follows, section I gives a brief and concise introduction to signal modeling and its associated challenges. In section II, some of the variations or types of modeling will be discussed while section III gives brief introduction to various methods of estimating model coefficients. In section IV, the detail of using Neural network reported in [7], [8] will be discussed. Section V will discuss the result obtained while the conclusion is as presented in section VI.

II. SIGNAL MODELING TYPES

Some of the known modeling methods include:

- **Autoregressive modeling Technique (AR)**: Consider a system describe by a linear constant coefficient difference equation (LCCDE) given by (1), the output $y(n)$ is obtained by using only previous outputs i.e $y(n-1)$, $y(n-2)$, $y(n-3)$ $\ldots y(n-p)$and the current input i.e $x(n)$, which means that $b(k)=0$ for k >0 and only $a(k)$ and $b(0)$ must be determined, such a system are called **Autoregressive (AR)** model.
  AR model equation is given

$$y(n)=-\sum_{k=1}^{p}a_k y(n-k)+b(0)x(n) \qquad (1)$$

and the equivalent z-domain is

$$Y(z)=-\sum_{k=1}^{p}a_k Y(z)z^{-k}+b(0)X(z) \qquad (2)$$

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b(0)}{1+\sum_{k=1}^{p}a_k z^{-k}} \qquad (3)$$

- **Autoregressive with external input(s) modeling Technique (ARX)**: From (4), the output $y(n)$ is given by

$$y(n)=-\sum_{k=1}^{p}a_k y(n-k)+b(0)x(n) \qquad (4)$$

adding an external input yields

International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + b(0)x(n) + \sum_{k=1}^{r} c_k u(n-k) \quad (5)$$

taking z-transform of both sides results in

$$Y(z) = -\sum_{k=1}^{p} a_k Y(z) z^{-k} + b(0)X(z) + \sum_{k=1}^{r} c_k U(z) z^{-k} \quad (6)$$

- **Autoregressive moving Average (ARMA) modeling technique**
  The general ARMA equation is given by

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) \quad (7)$$

taking z-transform of both sides results in

$$Y(z) = -\sum_{k=1}^{p} a_k Y(z) z^{-k} + \sum_{k=0}^{q} b_k X(z) z^{-k} \quad (8)$$

so that

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \quad (9)$$

in which

$$X(z) = \sum_{k=-\infty}^{\infty} x(n) z^{-n} \quad (10)$$

and

$$Y(z) = \sum_{k=-\infty}^{\infty} y(n) z^{-n} \quad (11)$$

- **Autoregressive moving Average with external input(s) (ARMAX) modeling technique**
  The general ARMA equation is given by 12,i.e

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) \quad (12)$$

the addition of external input to the system results in

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) + \sum_{k=1}^{r} c_k u(n-k) \quad (13)$$

taking z-transform of both sides results in

$$Y(z) = -\sum_{k=1}^{p} a_k Y(z) z^{-k} + \sum_{k=0}^{q} b_k X(z) z^{-k} + \sum_{k=1}^{r} c_k U(z) z^{-k} \quad (14)$$

other known modeling techniques include

- **Autoregressive Integrated Moving Model (ARIMA)**
- **Autoregressive Integrated Moving with external input Model (ARIMAX)**
- **Autoregressive Fractionally Integrated Moving Model (AFRIMA)**

In this report, The general ARMA given by eqn. 12 will be discussed and the relationship between AR, MA and ARMA modeling techniques is as contained in the Wold decomposition theorem [12]. This theorem shows that any stationary ARMA or MA process of finite variance can be represented as a unique AR process of possibly infinite order; likewise any ARMA or AR process can be represented as a MA process of possibly infinite order [12].

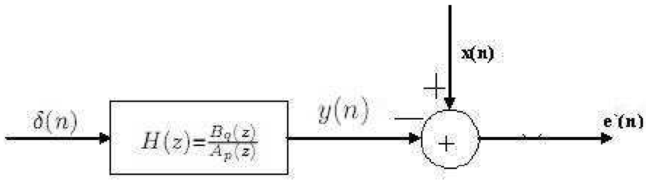

Fig. 1. Direct Method of least square method of ARMA Model

III. Methods of Coefficients determination

Various methods have been reported in literatures for determining the AR/ARMA model coefficients, among which are:

A. Direct least square method

The block diagram for direct method of least square solution is as shown in fig. 1 The modeling error can be written as

$$e(n) = x(n) - h(n)$$

in Frequency domain, we have

$$E(e^{j\omega}) = X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})} \quad (15)$$

In this method, the signal modeling to be minimized is the squared error,

$$\xi_{ls} = \sum_{n=0}^{\infty} |e(n)|^2$$

A necessary condition for the coefficients $a_p(k)$ and $b_q(k)$ to minimize the error is that the partial derivative of ξ_{ls} with respect to each of the coefficients vanishes. i.e

$$\frac{\delta \xi_{ls}}{\delta a_p^*(k)} = 0; \qquad k = 1, 2, \ldots, p$$

$$\frac{\delta \xi_{ls}}{\delta b_p^*(k)} = 0; \qquad k = 1, 2, \ldots, q$$

Using Parseval's theorem and taking the fourier transform of the error $e(n)$, we have

$$\xi_{ls} = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 d\omega \quad (16)$$

$$\frac{\delta \xi_{ls}}{\delta a_p^*(k)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\delta}{\delta a_p^*(k)} [E(e^{j\omega}) E^*(e^{j\omega})] d\omega = 0 \quad (17)$$

substituting (15) to (17), gives

$$\frac{\delta \xi_{ls}}{\delta a_p^*(k)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} [X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}] \frac{B_q^*(e^{j\omega})}{[A_p^*(e^{j\omega})]^2} e^{jk\omega} d\omega = 0 \quad (18)$$

and

International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467

$$\frac{\delta\xi_{ls}}{\delta b_q^*(k)} = -\frac{1}{2\pi}\int_{-\pi}^{\pi}[X(e^{j\omega}) - \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}]\frac{e^{jk\omega}}{A_p^*(e^{j\omega})}d\omega = 0 \tag{19}$$

B. Pade Approximation

The Pade approximation can be developed using fig. 1. The system function is

$$H(z) = \frac{B_q(z)}{A_q(z)} = \frac{\sum_{k=0}^{q} b_q z^{-k}}{1 + \sum_{k=1}^{p} a_p z^{-k}}$$

which leads to the difference equation

$$h(n) + \sum_{k=1}^{q} a_p(k)h(n-k) = b_q(n) \tag{20}$$

setting $h(n) = x(n)$ for $n = 0, 1, 2 \ldots, p+q$ in (20) yields a set of $p+q+1$ linear equations in $p+q+1$ unknowns, given by

$$h(n)+\sum_{k=1}^{q} a_p(k)h(n-k) = \left\{ \begin{array}{cc} b_q(n) & n = 1, 2, \ldots, q \\ 0 & n = q+1, \ldots q+p \end{array} \right\} \tag{21}$$

C. Prony method

Multiplying both sides of (15) by $A_p(z)$ yields

$$E(z) = A_p E'(z) - B_q(z)$$

in time domain,

$$e(n) = a_p(n) * x(n) - b_q(n) = \bar{b}_q(n) - b_q(n), \tag{22}$$

since $b_q(n) = 0$ for $n > q$, then,

$$e(n) = \left\{ \begin{array}{cc} x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) - b_q(n) & n = 1, 2, \ldots, q \\ x(n) + \sum_{l=1}^{p} a_p(l)x(n-l) & n = n > q \end{array} \right\} \tag{23}$$

$$\xi_{ls} = \sum_{n=0}^{\infty} |e(n)|^2 = \sum_{n=0}^{\infty} |x(n) + \sum_{l=1}^{p} a_p(l)x(n-l)|^2 \tag{24}$$

setting the partial derivatives of ξ_{ls} with respect to $a_p^*(k)$ equal to zero, gives

$$\frac{\delta\xi_{p,q}}{\delta a_p^*(k)} = \sum_{n=q+1}^{\infty} \frac{\delta[e(n)e^*(n)]}{\delta a_p^*(k)} = \sum_{n=q+1}^{\infty} e(n)\frac{\delta e^*(n)}{\delta a_p^*(k)} = 0 \tag{25}$$

$$\frac{\delta\xi_{p,q}}{\delta a_p^*(k)} = \sum_{n=q+1}^{\infty} e(n)x^*(n-k) = 0 \qquad k = 1, 2, \ldots p \tag{26}$$

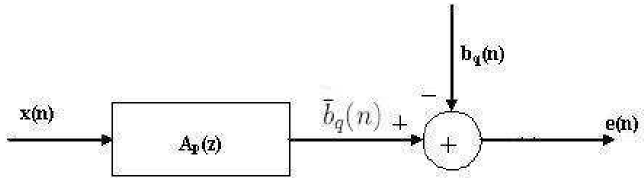

Fig. 2. Method of Prony Method

substituting eqn. (23) in eqn. (26),

$$\sum_{n=q+1}^{\infty} (x(n) + \sum_{l=1}^{p} a_p(l)x(n-l))x^*(n-k) = 0 \tag{27}$$

or equivalently,

$$\sum_{l=1}^{p} a_p(l)[\sum_{n=q+1}^{\infty} (x(n-l)x^*(n-k)] = -\sum_{n=q+1}^{\infty} (x(n)x^*(n-k)] \tag{28}$$

becomes

$$\sum_{l=1}^{p} a_p(l)r_x(k,l) = -r_x(k,0); \qquad k = 1, 2 \ldots, p \tag{29}$$

where

$$r_x(k,l) = \sum_{n=q+1}^{\infty} (x(n-l)x^*(n-k)$$

D. Shank method (Modified Prony)

Shank method is a modified Prony method in the sense that the moving average coefficients is obtained by finding the least square minimization of the model error over the entire data length [9]. The model can be viewed as a cascade of two filters, $B_q(z)$ and $A_p(z)$

The combine transfer function is given by,

$$H(z) = B_q(z)\frac{1}{A_p(z)}$$

The output of $y(n)$ can be computed using

$$g(n) = \delta(n) - \sum_{l=1}^{p} a_p(k)g(n-k)$$

The numerator coefficient is obtained by minimizing the square of the error.

$$\xi_S = \sum_{n=0}^{\infty} |e(n)|^2$$

minimizing the error in Prony method gives

$$\sum_{l=0}^{q} b_p(l)r_y(k,l) = r_{xy}(k); \qquad k = 1, 2 \ldots, q \tag{30}$$

where

$$r_y(k-l) = \sum_{n=0}^{\infty}(y(n-l)y^*(n-k)$$

and

$$r_{xy}(k) = \sum_{n=0}^{\infty}(x(n)y^*(n-k)$$

The block diagram for Shank's method is as shown in fig. 3

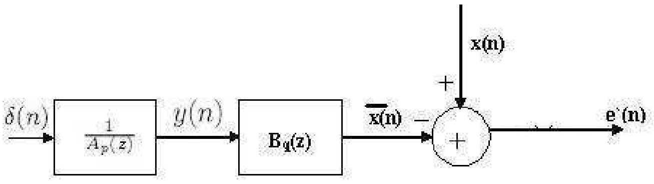

Fig. 3. Method of Shank method

E. Autocorrelation method

In this method, the signal $x(n)$ is only known over a finite data sequence $[0, N]$, this is obtained by multiplying an infinite data sequence by a window function $w(n)$ to obtain another signal $x'(n)$. i.e

$$x'(n) = \left\{ \begin{array}{cc} s(n)w(n) & 0 \leq n \leq N-1 \\ 0 & otherwise \end{array} \right\} \quad (31)$$

Using Prony's method to find an all pole model for $x'(n)$ by minimizing $a_p(k)$ coefficients as stated in (26), (28), (29), yields

$$\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0); \qquad k = 1, 2 \ldots, p \quad (32)$$

where

$$r_x(k) = \sum_{n=k}^{N} x(n)x^*(n-k)$$

The autocorrelation matrix formed is a symmetric Toeplitz matrix.

F. Covariance method

In contrast to Autocorrelation method discussed in section III-E, the error in (24) is minimized over a definite interval $[p, N]$, that is

$$\xi_p^C = \sum_{n=p}^{N} |e(n)|^2 \quad (33)$$

The only difference between covariance method and Prony method is in the summation of the error term [9], the covariance normal equations can be written as

$$\sum_{l=1}^{p} a_p(l) r_x(k,l) = -r_x(k,0); \qquad k = 1, 2 \ldots, p \quad (34)$$

where

$$r_x(k,l) = \sum_{n=p}^{N}(x(n-l)x^*(n-k)$$

IV. AR/ARMA Coefficients determination using Artificial Neural Network Technique

The general three layer neural network of obtaining the coefficients of ARMA and NARMA reported in [7], [8] is as shown in figure 4

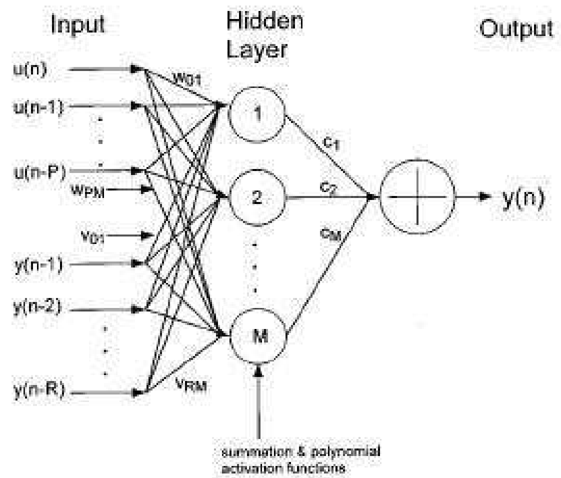

Fig. 4. Neural Network Technique for obtaining ARMA/NARMA coefficients. [7], [8]

For a system define by,

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) \quad (35)$$

the coefficients are obtained from the neural network weights value and polynomial coefficients given by eq. 54 and eq. 55.

$$a_i = \sum_{j=1}^{M} w_{j1} a_{1j} v_{ij} y(n-i) \quad (36)$$

$$b_i = \sum_{j=1}^{M} w_{j1} a_{1j} v_{ij} x(n-i) \quad (37)$$

A. ARMA-ANN Coefficients determination Algorithm

1) **Stopping criteria:**
Set the stopping criteria; Epoch or Mean Square Error (MSE)
2) **Initialize:**
Number of input nodes equal model order. i.e (Input nodes = $p + q$)
Output node is one(1), $y(n)$
Initialize the weight vectors
Intialize Polynomial Order, R (R=2)
Intialize all Polynomial coefficients,a_{zi} where

$$P_i(t) = \sum_{z=0}^{R} a_{zi} t^z \quad (38)$$

3) **Training Pattern :**
Select the training input and output pairs for the network.

International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467

International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467

Input pattern, AR section: $y(n-1), y(n-2), \ldots y(n-p)$
Input pattern, MA section: $x(n), x(n-1), \ldots x(n-q)$
Target pattern, $T(n)$

4) **Run selected pattern**
The output node $y(n)$ is given by,

$$y(n) = \sum_{k=1}^{M} w_{k1} h_i(t) \quad (39)$$

and the hidden nodes output is given as,

$$h_i(t) = \sum_{z=0}^{R} a_{zi} t^z \quad (40)$$

where the AR part is given by

$$t(n) = \sum_{j=1}^{p} v_{k1} y_i(n-j) + \phi(j) \quad (41)$$

5) **Evaluate the Error**
The output Error is

$$\delta(n) = T(n) - y(n) \quad (42)$$

6) **Back Propagate the Error**

$$\delta_k(n) = w_k * \delta(n) \quad (43)$$

7) **Weight Update (Input to Hidden layer)**

$$v_{jk(new)} = v_{jk(old)} + \eta \delta_k \frac{\delta P_k(t)}{\delta t} y(n-j) \quad (44)$$

where

$$\frac{\delta P_k}{\delta t} = \sum_{b=1}^{2} b a_{bk} t^{(b-1)} \quad (45)$$

8) **Polynomial Coefficient Update (Hidden layer)**

$$a_{jknew} = a_{jkold} + \eta \delta_1 \frac{\delta P_i}{\delta a_{jk}} \quad (46)$$

where

$$\frac{\delta P_i}{\delta a_{0i}} = 1 \quad (47)$$

$$\frac{\delta P_i}{\delta a_{1i}} = t_i \quad (48)$$

$$\frac{\delta P_i}{\delta a_{2i}} = t_i^2 \quad (49)$$

(50)

which can be simply put as

$$\frac{\delta P_i}{\delta a_{ri}} = t_i^r \quad (51)$$

for $r = 0, 1$ and 2 , i is the hidden layer number and t_i is as defined in 41 for the AR section.

9) **Weight Update (Hidden layer to Output Layer)**

$$w_{k1new} = w_{k1old} + \eta \delta \frac{\delta P_k}{\delta t} P_k \quad (52)$$

For Output neuron, the activation function is $P(t) = t$ so the derivative $\frac{\delta P_1}{\delta t} = \frac{\delta P_2}{\delta t} = 1$.
So,

$$w_{k(new)} = w_{k(old)} + \eta \delta P_k \quad (53)$$

10) **Test for Stopping criteria:**
Test for stopping criteria.
If completion criteria is not satisfied, go to step 4 else calculate and output the ARMA coefficients

$$a_i = \sum_{j=1}^{M} w_{j1} a_{1j} v_{ij} y(n-j) \quad (54)$$

$$b_i = \sum_{j=1}^{M} w_{j1} a_{1j} v_{ij} x(n-j) \quad (55)$$

11) Ends

V. Result obtained

In this paper, results obtained by the use of Pade, Prony Shank, Autocorrelation and Covariance method coefficients determination types will be reported.

1) Autoregressive Equation 1

$$y(n) = y(n-1) + 0.24y(n-2) + w(n) \quad (56)$$

where w(n) is white noise.

2) Autoregressive Equation 2

$$\begin{aligned} y(n) \quad = \quad & 0.51y(n-1) + 0.315y(n-2) - 0.23y(n-3) \\ & -0.56y(n-4) + 0.1y(n-5) - 0.045y(n-6) + w(n) \end{aligned}$$

where w(n) is white noise.

3) Autoregressive Moving Average

$$\begin{aligned} y(n) \quad = \quad & 0.11y(n-1) + 0.52y(n-2) \\ & +x(n) - 0.3x(n-1) - 0.078x(n-2) \end{aligned}$$

TABLE I
Result obtained from eqn. 56

Methods	$a(1)$	$a(2)$
Actual Value	1.000	0.240
Pade	0.670	0.203
Prony	0.833	0.215
Shank	0.872	0.233
Autocorrelation	0.903	0.240
Autocovariance	1.001	0.240
NN	1.000	0.241

TABLE II
Result obtained from eqn. 57

Methods	$a(1)$	$a(2)$	$a(3)$	$a(4)$	**a(5)**	**a(6)**
Actual Value	0.510	0.315	−0.230	−0.560	0.100	−0.045
Pade	0.704	0.033	−0.202	−0.167	0.488	−0.279
Prony	0.504	0.331	−0.202	−0.667	0.188	−0.178
Shank	0.513	0.331	−0.202	−0.668	0.103	−0.179
Autocorrelation	0.530	0.285	−0.202	−0.667	0.089	−0.179
Autocovariance	0.510	0.322	−0.230	−0.561	0.104	−0.046
NN	0.510	0.314	−0.230	−0.560	0.102	−0.045

TABLE III
RESULT OBTAINED FROM EQN. 57

Methods	a(1)	a(2)	b(1)	b(2)	b(3)
Actual Value	0.110	−0.500	0.411	−0.390	−0.685
Pade	0.421	−0.341	0.223	−0.219	−0.355
Prony	0.221	−0.600	0.366	−0.431	−0.534
Shank	0.221	−0.5980	0.380	−0.439	−0.534
Autocorrelation	0.121	−0.631	0.372	−0.343	−0.639
Autocovariance	0.111	−0.500	0.411	−0.390	−0.680
NN	0.110	−0.500	0.411	−0.390	−0.685

VI. CONCLUSION

In this paper, different methods of determining ARMA coefficients have been evaluated based on a simulated data. An algorithm in achieving the reported methods in [7], [8] have also been discussed. MATLAB implementation of this have also been carried out . This work only consider the accuracy of the coefficients and not the time of completion of each of the method. Result obtained shows that this method efficiently and accurately compute ARMA coefficients. Result obtained also shows that the result perform better than some of the existing method of ARMA coefficient beacuse of the non linearity nature of ANN. Area of application of this proposed algorithm include Magnetic Resonance Imaging reconstruction using parametric technique [4], Signal modeling, Adaptive control system and PID tunning.

Acknowledgement: We shall like to express our appreciation to Prof. K. H. Chon for all his suggestion, advice and contribution to this work.

REFERENCES

[1] Z. P. Liang, P. C. Lauterbur, “Principles of Magnetic Resonance Imaging, A signal processing perspective”, IEEE Press, New York, 2000.

[2] A. M. Aibinu, M. J.E. Salami, A. A. Shafie and A. R. Najeeb, “ Model Order Determination for MRI Signal”, *accepted for publication* , 2008.

[3] D. G. Nishimura, “Principles of Magnetic Resonance Imaging”, April 1996.

[4] M. R. Smith, S. T. Nichols, R. M. Henkelman and M. L. Wood, “Application of Autoregressive Moving Average Parametric Modeling in Magnetic Resonance Image Reconstruction”, *IEEE Transactions on Medical Imaging*, Vol. M1-5:3, pp 257 - 261, 1986.

[5] F. J. Harris,“On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform”, Proceedings of the IEEE. Vol. 66, January 1978.

[6] M. R. Smith, S. T. Nichols, R. Constable and R. Henkelman, “A quantitative comparison of the TERA modeling and DFT magnetic resonance image reconstruction techniques”, *Magn. Reson. Med.*, Vol. 19 pp. 1-19, 1991.

[7] K. H. Chon, R. J. Cohen, “Linear and Non-Linear ARMA Model Parameter Estimation Using Artificial Neural Network”, *IEEE Transactions on BioMedical Engineering*, Vol. 44, No 3, pp 168 - 174, 1997.

[8] K. H. Chon, D. Hoyer, A. A Armoundas, N-H Holstein-Rathlou and D. J Marsh, “ Robust Nonlinear Autoregressive Moving Average Model Parameter Estimation Using Recurrent Artificial Neural Network”, *Annals of BioMedical Engineering*, Vol. 27, pp 538-547, 1999.

[9] M. H. Hayes, “Staitical Digital Signal processing and Modelling”, John Wiley & Sons, Canada, 1996.

[10] Z. P. Liang, F. E. Boada, R. T. Constable, E. M. Haacke, P. C. Lauterbur, and M. R. Smith, “Constrained Reconstruction Methods in MR Imaging”, Reviews of MRM, vol. 4, pp.67 - 185, 1992.

[11] E. Hackle and Z. Liang, “Superresolution Reconstruction Through Object Modeling and Estimation”, *IEEE transactions in A.S.S.P*, 37: 592 - 595, 1989.

[12] E. C. Whitman, “The spectral analysis of discrete time series in terms of linear regressive models”, *Naval Ordinance Labs Rep.*, NOLTR-070-109, White Oak, MD, June 23, 1974.

[13] R. Palaniappan, “Towards Optimal Model Oreder Selection for Autoregressive Spectral Analysis of Mental Tasks Using Genetic Algorithm”, *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6 No. 1A, January 2006.

[14] H. Akaike, “Power Spectrum Estimation through Autoregression Model Fitting”, *Annals of the Institute of Statistical Mathematics*, vol. 21, pp. 407-419, 1969.

[15] H. Akaike, “A New Look at the Statistical Model Identification”, IEEE Trans. Autom. Control, vol. AC-19, pp. 716-723, 1974.

[16] J. Rissanen, ”Modelling by shortest data description”, Automatica, vol.14, pp.465-471, 1978.

[17] M.J. Salami, A. R. Najeeb, O. Khalifa, K. Arrifin, “MR Reconsturction with Autoregressive Moving Average”, *International Conference on Biotechnology Engineering*, Kuala Lumpur, pp 676 - 704, May, 2007.

International Science Index, Computer and Information Engineering Vol:2, No:6, 2008 waset.org/Publication/1467