Identification of Multivariable Wiener Model Using Radial Basis Functions Neural Networks

Syed Saad Azhar Ali, *Member, IEEE* and Hussain N. Al-Duwaish

***Abstract*—A new method is introduced for the identification of the nonlinear multi input multi output (MIMO) Wiener Model, comprising of linear dynamics in cascade with static nonlinearities. The static nonlinearities are modeled by Radial Basis Function Neural Networks (RBFNN) and the linear part is modeled by MIMO autoregressive moving average (ARMA) model. The new algorithm makes use of the well known mapping ability of RBFNN. The learning algorithm is an extension of SISO identification scheme presented in [1] . The proposed algorithm estimates the weights of the RBFNN and the coefficients of ARMA model based on least mean squares (LMS) principle simultaneously.**

***Index Terms*—Wiener model, multivariable, Radial Basis Function Neural Network.**

I. Introduction

The behavior of many systems can be approximated by a dynamic linear part in cascade with a static nonlinearity. This type of model is known as Wiener Model. Wiener Model

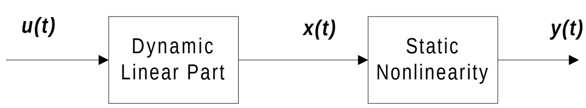

Fig. 1. Structure of Wiener Model

comprises of two blocks in cascade shown in figure 1, the first one a non-linear and the second one a linear. The Wiener model is used to model several classes of nonlinear systems. Its flexibility lies in having the Nonlinearity entirely separate from the common and easily realizable linear parts. Thus the Non-linearity can be identified by considering it separately. This Identification gives more insight into the system and makes the system more observable and simplifies the Controller design.

The examples in which such a system was applied are Nonlinear filters [2], Nonlinear networks [3], Detection of Signals in Non-Gaussian Noise [5],Nonlinear Prediction [6], Nonlinear Data transmision channels [7], Control Systems [8], Identification of Nonlinear Systems [9], Biological systems [10] and many others.

Extensive research has been performed over the years. Billings and Fakhouri [11] presented a cross correlation technique to identify the Wiener model. The estimate of all the linear parts of the system were achieved by using cross correlation technique and using these estimates, the outputs were generated for all the systems. The output error was used to update the estimate and then it was iterated until the desired outputs were generated.

Hu and Wang [12] used a three level pseudo-random sequence to estimate the impulse responses of the linear part and the polynomial coefficients of the nonlinearity of the discrete Wiener model.

Duwaish *et al.* [13] presented a neural network approach towards the identification of the Wiener model. The linear part was modelled by an ARMA, where the coefficients were updated using the RLS, and the nonlinearity was estimated by MFNN. The MFNN is updated using the back-propagation algorithm.

A recursive subspace identification algorithm was proposed by Lovera and Verhaegen [14]. The method was found as a by product of the MOESP (multivariable output error state-space model identification) class algorithm.

Chou and Verhaegen [15] used the pre-filtering of both input and output data. This technique is reported to have good results for LTI systems. The authors showed that correlation method can be used for the nonlinear systems to apply data pre-filtering.

Similar to Hammerstein systems, the Wiener model was also identified using the maximum likelihood with a linear regression initialization by Hagenblad and Ljung [16]. The idea is to first parameterize the model, and then parameters are transformed in the form of original system. The inverse of the nonlinear system is parameterized with linear B-splines. Now the estimate of the intermediate signal from the parameters and the B-splines are equated and solved for the parameters.

Rodriguez and Fleming [17] used a multi-objective approach in the genetic programming to identify the nonlinearity in the Wiener model. This method gives a tradeoff between the complexity and the performance of the system.

Greblicki [18] used orthogonal series expansion for the identification. The author also used Kernels regressions in [19] to identify the Wiener model. Greblicki also identified continuous time Hammerstein system [20] and continuous time Wiener system [21].

Hagenblad used the prediction error and expectation maximization method to identify the Wiener model in [22].

Chou and Verhaegen [23] adopted a three step algorithm to identify the Wiener model with process noise. The algorithm was based on cross-correlation and subspace identification. The subspace matrices are estimated first which are used in the estimation of the intermediate signal and together with

Syed Saad Azhar Ali is with the Faculty of Engineering, Sciences and Technology, Iqra Univeristy, Karachi, Pakistan. e-mail: saadazhar@iqra.edu.pk.
Hussain N. Al-Duwaish is with King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia. email: hduwaish@kfupm.edu.sa.

the output the inverse of the static nonlinearity is estimated. Using this estimated nonlinearity the consistent estimates of the subspace variables are approached.

Yong and Chow [24] presented a hybrid model of wavelets and neural networks as an identification scheme for the Wiener model. The orthogonal scaling and the mother wavelets were combined to obtain the activation functions. The neural network is used to avoid the need of any $apriori$ information. The wavelets capture the modes of the nonlinearity by passing the nonlinearity through a time-frequency plane window. The liner part was identified by an ARMA.

A comprehensive analysis of stochastic gradient identification of Wiener systems was done by Celka *et al.* [25].

Anders and Lars [26] identified the time varying Wiener Hammerstein system. They made the use of the extended Kalman filter. Furthermore, to ensure stability, the paper reformulated the algorithm in terms of a nonlinear minimization problem with a quadratic inequality constraint.

Lacy *et al.* [27] identified the Wiener model by minimizing a cost function for standard least squares depending upon the vector of unknown system parameters and the intermediate signal.

The algorithm presented in [13] used multi-layered feed forward neural networks (MFNN) to identify the static nonlinearities in the Wiener model. The back-propagation algorithm is used to update the weights of the MFNN. It is well known in the literature that the convergence of the back-propagation algorithm is very slow compared to LMS when used for training radial basis functions neural networks (RBFNN) [28], [29], [30], [31]. Moreover, the RBFNN has the same universal approximation capabilities as the MFNN [28]. This motivated the use of RBFNN instead of MFNN to model the static nonlinearity. The linear part is modeled by an ARMA model.

II. MIMO Wiener Systems

MIMO nonlinear systems are very common in practice for example distillation column, chemical and biological processes. A MIMO Wiener system can be classified into two classes with respect to the type of static nonlinearities in the system. The static nonlinearities can be separate or combined. In this paper systems with separate nonlinear parts are consider.

A. MIMO Wiener System with Separate Nonlinearities

Consider a general MIMO Wiener system with M-input, M-intermediate variables and N-output shown in Fig. 6. All the static nonlinearities are separate and each can be considered as an independent SISO nonlinear block. So, for M nonlinearities at the input, there are M inputs and M intermediate variables.

The M inputs given by $U(t) = [u_1(t)\ u_2(t)\ \ldots\ u_M(t)]^T$ are fed to M static nonlinearities modeled by RBFNN. The outputs of the static nonlinearities are:

$$X(t) = [x_1(t)\ x_2(t)\ \ldots\ x_M(t)]^T, \tag{1}$$

and $x_i(t)$ estimated by the RBFNN is given by,

$$\hat{x}_i(t) = W_i \phi(\|u_i(t) - C_i\|). \tag{2}$$

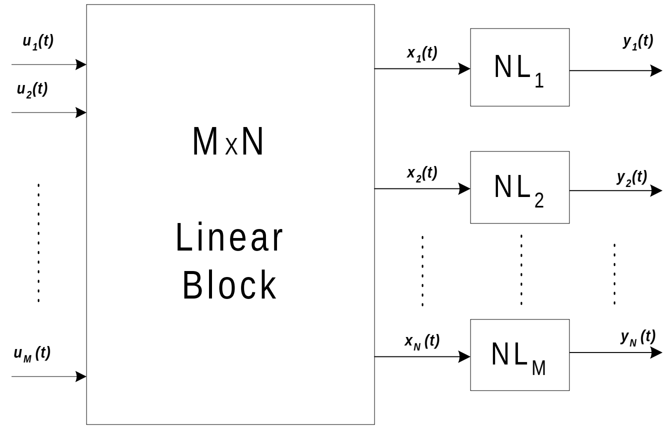

Fig. 2. An M-input N-output Wiener system with separate nonlinearities.

The system output can be defined by:

$$Y(t) = [y_1(t)\ y_2(t)\ \ldots\ y_N(t)]^T, \tag{3}$$

which is

$$Y(t) = \begin{bmatrix} G_{11} & G_{12} & \ldots & G_{1M} \\ G_{21} & G_{22} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & \ldots & \ldots & G_{NM} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix}. \tag{4}$$

The estimated outputs can be written in a similar fashion shown in algebraic form as:

$$\begin{aligned} \hat{y}_1(t) &= \hat{x}_1(t)\hat{G}_{11} + \hat{x}_2(t)\hat{G}_{12} + \ldots + \hat{x}_M(t)\hat{G}_{1M}, \\ \hat{y}_2(t) &= \hat{x}_1(t)\hat{G}_{21} + \hat{x}_2(t)\hat{G}_{22} + \ldots + \hat{x}_M(t)\hat{G}_{2M}, \\ &\vdots \\ \hat{y}_N(t) &= \hat{x}_1(t)\hat{G}_{N1} + \hat{x}_2(t)\hat{G}_{N2} + \ldots + \hat{x}_M(t)\hat{G}_{NM}, \end{aligned}$$

where the transfer function corresponding to the j^{th} intermediate variable and i^{th} output is,

$$\hat{G}_{ij} = \frac{B^w_{ij}(q^{-1})}{A(q^{-1})}. \tag{5}$$

The polynomials $A(q^{-1})$ and $B^w_{ij}(q^{-1})$ are defined as,

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_n q^{-n}, \\ B^w_{ij}(q^{-1}) &= b_{0ij} + b_{1ij} q^{-1} + \cdots + b_{wij} q^{-w}, \end{aligned}$$

where w is the order of zeros of that particular transfer function.

III. RBFNN/ARMA Identification Scheme for MIMO Wiener System

Identification structure composed of RBFNN and ARMA in series has successfully been used to identify SISO Wiener Model in [32]. Following the same idea, a nonlinear system with more than one inputs and outputs can also be identified with MIMO RBFNN/ARMA structure.

A. Radial Basis Function Neural Networks

RBFNN is a type of feedforward neural network. They are used in a wide variety of contexts such as function approximation, pattern recognition and time series prediction. Networks of this type have the universal approximation property. In these networks the learning involves only one layer with lesser computations. This results in reduction in the training time in comparison with multi layered feedforward neural networks (MFNN), that use back propagation algorithm to update the weights of all the layers. These features make RBFNN attractive in many practical problems. A SISO RBFNN is shown

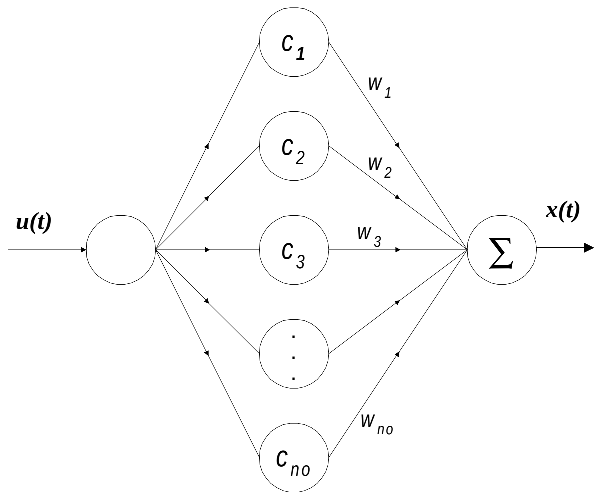

Fig. 3. A general SISO RBF network

in Fig. 3. It consists of an input node $x(t)$, a hidden layer with n_o neurons and an output node $y(t)$. Each of the input node is connected to all the nodes in the hidden layer through unity weights (direct connection). While each of the hidden layer nodes is connected to the output node through some weights $w_1, \dots, w_{n_o}$. Each neuron finds the distance, normally applying Euclidean norm, between the input and its center and passes the resulting scalar through a non-linearity. So the output of the hidden neuron is given by $\phi(\|x(t) - c_i\|)$, where n_o is the number of hidden layer nodes (neuron), $x(t)$ is the input, c_i is the center of i^{th} hidden layer node where $i = 1, 2, \dots, n_o$, and $\phi(\cdot)$ is the nonlinear basis function. Normally this function is taken as a Gaussian function of width β. The output $(y(t))$ is a weighted sum of the outputs of the hidden layer, given by

$$\begin{aligned} y(t) &= W\Phi(t), \\ y(t) &= \sum_{i=1}^{n_o} w_i \phi(\|x(t) - c_i\|), \end{aligned}$$

$$\text{where} \quad W = [w_1 \; w_2 \; \dots \; w_{n_o}],$$

$$\text{and } \Phi(t) = [\|x(t) - c_1\| \; \|x(t) - c_2\| \; \dots \; \|x(t) - c_{n_o}\|]^T.$$

$y(t)$ is the output, w_i is the weight corresponding to the i^{th} hidden neuron. An M input P output RBFNN is shown in Fig. 4.

$$Y(t) = [y_1(t) y_2(t) \dots \; y_P(t)],$$

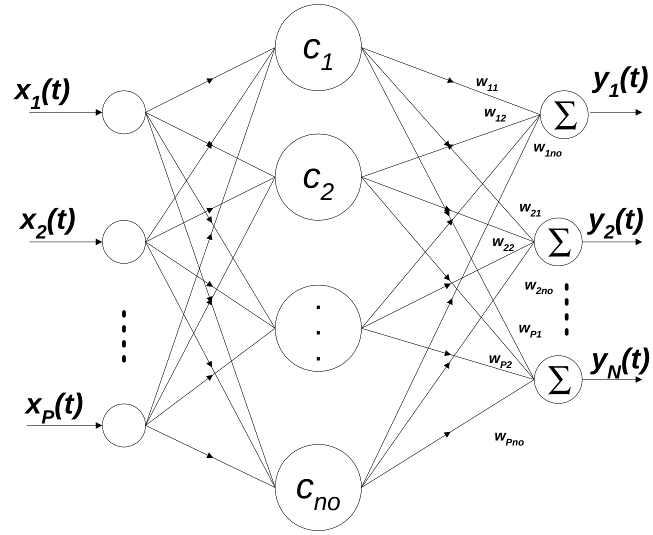

Fig. 4. A MIMO RBFNN network.

where the j^{th} output has the effect of all the M inputs and w_{ij} is the weight connecting the i^{th} neuron to the j^{th} output.

$$x_j(t) = \sum_{k=1}^{M} \sum_{i=1}^{n_o} w_{ij} \phi(\|x_k(t) - c_i\|).$$

B. ARMA Model

The linear part of the Wiener model is modeled by an ARMA model, whose output is given by

$$x(t) = \sum_{i=1}^{n} a_i x(t-i) + \sum_{j=0}^{m} b_j u(t-j) \tag{8}$$

or in terms of q^{-1} operator

$$x(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} u(t) \tag{9}$$

IV. Development of Training Algorithms

Considering Fig. 5, the objective is to develop a recursive algorithm by which the weights of the RBFNN and coefficients of the ARMA model can be adjusted, such that the set of inputs produces the desired set of outputs. This problem is solved by developing a new parameter estimation algorithm which will be based on the well known LMS principles.

The parameters (weights of RBFNN and the coefficients of ARMA) are updated by minimizing the performance index I given by,

$$\begin{aligned} I &= \frac{1}{2} E(t)^T E(t), \qquad (10) \\ E(t) &= Y(t) - \hat{Y}(t), \end{aligned}$$

where $Y(t)$ is the vector of actual outputs of the system and $\hat{Y}(t)$ is the vector of estimated outputs of the MIMO Wiener

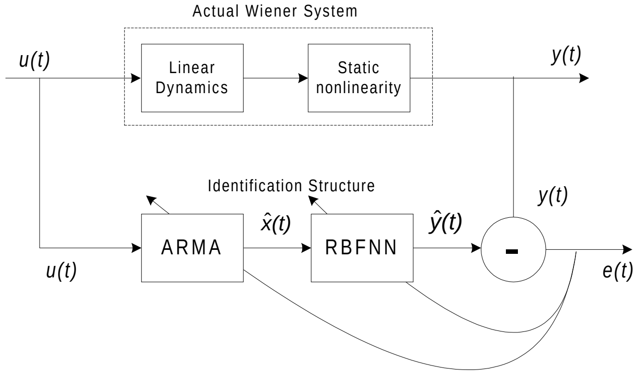

Fig. 5. Identification of Wiener system.

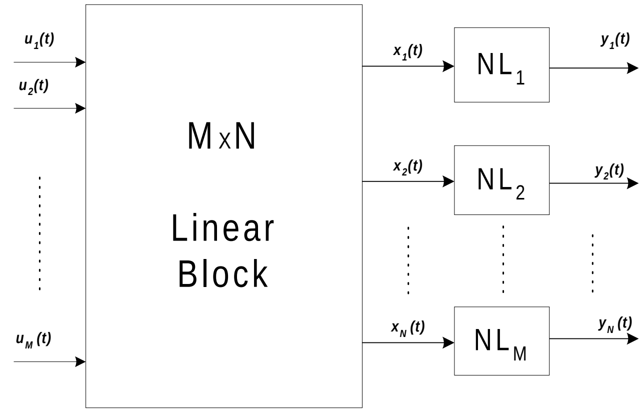

Fig. 6. An M-input N-output Wiener system with separate nonlinearities.

Model. Therefore the vector $E(t)$ is defined as:

$$\begin{aligned} E(t) &= [e_1(t)\ e_2(t)\ \ldots\ e_N(t)]^T, \\ \text{where} \quad e_1(t) &= y_1(t) - \hat{y}_1(t), \\ &\vdots \\ e_N(t) &= y_N(t) - \hat{y}_N(t), \\ \therefore \quad E(t) &= [(y_1(t) - \hat{y}_1(t))\ \ (y_2(t) - \hat{y}_2(t))\ \ldots \\ &\qquad \ldots\ (y_N(t) - \hat{y}_N(t))]^T. \end{aligned}$$

The coefficients of the ARMA model and the weights of the RBFNN should be updated in the negative direction of the gradient as,

$$\theta(K+1) = \theta(K) - \alpha \frac{\partial I}{\partial \theta(K)}, \tag{11}$$

and

$$W(K+1) = W(K) - \alpha \frac{\partial I}{\partial W(K)}, \tag{12}$$

where θ is the parameter vector, W is the weight vector for RBFNN and α is the learning parameter. The variable K is used to show the iteration number of training. This must not be confused with the time variable t.

Consider M-input N-output system shown in Fig. 6. There are N separate static nonlinearities at the output, fed by the same number of intermediate variables. The N intermediate variables are obtained by feeding the M inputs to the linear dynamic block.

The number of intermediate variables $x(t)$ is the same as the number of outputs of the system, since, there are as much nonlinearities as the number of intermediate variables.

$$X(t) = [x_1(t)\ x_2(t)\ \ldots\ x_N(t)]^T, \tag{13}$$

defined by,

$$X(t) = \begin{bmatrix} G_{11} & G_{12} & \ldots & G_{1M} \\ G_{21} & G_{22} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & \ldots & \ldots & G_{NM} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_M(t) \end{bmatrix} \tag{14}$$

The system outputs are given by $Y(t) = [y_1(t)\ \ y_2(t)\ \ \ldots\ \ y_N(t)]$, where the estimate of $y_j(t)$ is given by,

$$\hat{y}_j(t) = W_j \phi(\|\hat{X}(t) - C_i\|).$$

The partial derivatives of the performance index given by Eq. 10 $w.r.t.$ the weights and coefficients are found as follows.

$$\begin{aligned} \frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}\big(E(t)^T E(t)\big), \qquad (15) \\ &= E^T(t)\frac{\partial}{\partial W}\big[(y_1(t) - \hat{y}_1(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))\big], \\ &= -E^T(t)\frac{\partial}{\partial W}\big[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\big]. \qquad (16) \end{aligned}$$

considering the derivative $w.r.t. W_i$ only,

$$\begin{aligned} \frac{\partial I}{\partial W_i} &= -E^T(t)\frac{\partial}{\partial W_i}\big[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\big], \\ &= -E^T(t)\frac{\partial}{\partial W_i}\big[W_1\phi_1(t)\ W_2\phi_2(t)\ \ldots\ W_i\phi_N(t)\big], \\ &= -e_i(t)\phi_i(t). \end{aligned}$$

Defining $\Phi(t) = [\phi_1(t)\ \phi_2(t)\ \ldots\ \phi_N(t)]^T$. Therefore, Eq. 16 becomes,

$$\frac{\partial I}{\partial W} = -E^T(t)\Phi(t) \tag{17}$$

Eq. 17, together with Eq. 12 gives the final weight update equation,

$$W(K+1) = W(K) + \alpha E^T(t)\Phi(t) \tag{18}$$

Now finding the derivatives with respect to the coefficients.

$$\begin{aligned} \frac{\partial I}{\partial \theta} &= \frac{1}{2}\frac{\partial}{\partial \theta}\big(E(t)^T E(t)\big), \\ &= E^T(t)\frac{\partial}{\partial \theta}\big[(y_1(t) - \hat{y}_1(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))\big]^T, \\ &= -E^T(t)\frac{\partial}{\partial \theta}\big[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\big]^T. \qquad (19) \end{aligned}$$

Considering only the partial derivative $w.r.t.\ a_i$,

$$\frac{\partial I}{\partial a_i} = -E^T(t)\frac{\partial}{\partial a_i}\big[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\big]^T, \tag{20}$$

and now differentiating only $\hat{y}_j$,

$$\frac{\partial \hat{y}_j}{\partial a_i} = \frac{\partial W_j \phi_j(t)}{\partial a_i}.$$

The above equation can be taken as the derivative for single output. The derivative for single output is derived in chapter 3. For the MIMO system for the j^{th} intermediate variable and j^{th} output, the derivative $w.r.t.\ a_i$ will be:

$$\frac{\partial \hat{y}_j}{\partial a_i} = -\frac{2}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi(\|\hat{x}_j(t)-c_{jl}\|). \quad (21)$$

Stacking from Eq. 21 in Eq. 20 to obtain,

$$\frac{\partial I}{\partial a_i} = -E^T(t)\times \quad (22)$$
$$\begin{bmatrix} -\frac{2}{\beta_1^2}\hat{x}_1(t-1)\sum_{l=1}^{n_1}(\hat{x}_1(t)-c_{1l})w_{1l}\phi(\|\hat{x}_1(t)-c_{1l}\|) \\ \vdots \\ -\frac{2}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi(\|\hat{x}_j(t)-c_{jl}\|) \\ \vdots \\ -\frac{2}{\beta_N^2}\hat{x}_N(t-1)\sum_{l=1}^{n_N}(\hat{x}_N(t)-c_{Nl})w_{Nl}\phi(\|\hat{x}_N(t)-c_{Nl}\|) \end{bmatrix}$$

or

$$\frac{\partial I}{\partial a_i} = \quad (23)$$
$$2\sum_{j=1}^{N}\left(\frac{e_j(t)}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi(\|\hat{x}_j(t)-c_{jl}\|)\right),$$

and the update equation for a_i using Eq. 11 and Eq. 23 will be,

$$a_i(K+1) = a_i(K) - \quad (24)$$
$$2\alpha\sum_{j=1}^{N}\left(\frac{e_j(t)}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi_j(\|\hat{x}_j(t)-c_{jl}\|)\right)$$

Similarly from the SISO case the, derivative $w.r.t.\ b_{ijk}\ i.e.\ b_i$ for k_{th} input and j_{th} intermediate variable will be,

$$\frac{\partial I}{\partial b_{ijk}} = \quad (25)$$
$$\frac{2e_j(t)}{\beta_j^2}\ u_k(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi(\|\hat{x}_j(t)-c_{jl}\|),$$

and the update equation using Eq. 11 and Eq. 25,

$$b_{ijk}(K+1) = b_{ijk}(K) - \quad (26)$$
$$\frac{2\alpha e_j(t)}{\beta_j^2}\ u_k(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t)-c_{jl})w_{jl}\phi(\|\hat{x}_j(t)-c_{jl}\|).$$

V. Simulation Results

A 2-input 2-output example is considered here to verify the training algorithm for MIMO Wiener systems. One of the static nonlinearities is of saturation nonlinearity and the other one is a control valve. The characteristics of the saturation nonlinearity are defined as,

$$y_1(t) = \begin{cases} 0.5, & \text{for } x_1(t) > 0.5 \\ x_1(t), & \text{for } -0.5 \le x_1(t) \le 0.5 \\ -0.5, & \text{for } x_1(t) < -0.5 \end{cases} \quad (27)$$

The control valve is defined in [13] as,

$$y_2(t) = \frac{x_2(t)}{\sqrt{0.1+0.9x_2^2(t)}} \quad (28)$$

In this example third order linear systems are considered. The order of the poles is the same for all the transfer function in the linear block, this means all the outputs have the same number of delayed regressions.

$$\begin{aligned} y_1(t) &= 0.35y_1(t-1) - 0.65y_1(t-2) + 0.15y_1(t-3) \\ &\quad + 0.81x_1(t) - 0.53x_2(t), \\ y_2(t) &= 0.35y_2(t-1) - 0.65y_2(t-2) + 0.15y_2(t-3) \\ &\quad + 0.62x_1(t) - 0.22x_2(t). \end{aligned}$$

Using random numbers uniformly generated in the interval $[-2, 2]$, the desired outputs are produced by using the process model. The proposed identification scheme comprised of two RBFNNs in series with a MIMO ARMA model. The centers for both the RBFNNs are evenly located in the input space. The width of the basis function for the saturation nonlinearity is kept as 0.5 and for the heat exchanger as 0.6. The learning rate was 0.04. These values are selected after few trial runs and observing the SISO examples [1]. The linear part was modeled by the following MIMO ARMA model,

$$\begin{aligned} y_1(t) &= a_1y_1(t-1) + a_2y_1(t-2) + a_3y_1(t-2) \\ &\quad + b_{011}x_1(t) + b_{012}x_2(t), \\ y_2(t) &= a_1y_2(t-1) + a_2y_2(t-2) + a_3y_2(t-2) \\ &\quad + b_{021}x_1(t) + b_{022}x_2(t). \end{aligned}$$

The proposed algorithm is applied to update the parameters of the identification scheme. The RBFNNs identified the static nonlinearities very accurately. The actual and identified nonlinearities are shown in Fig. 7 and Fig. 8.

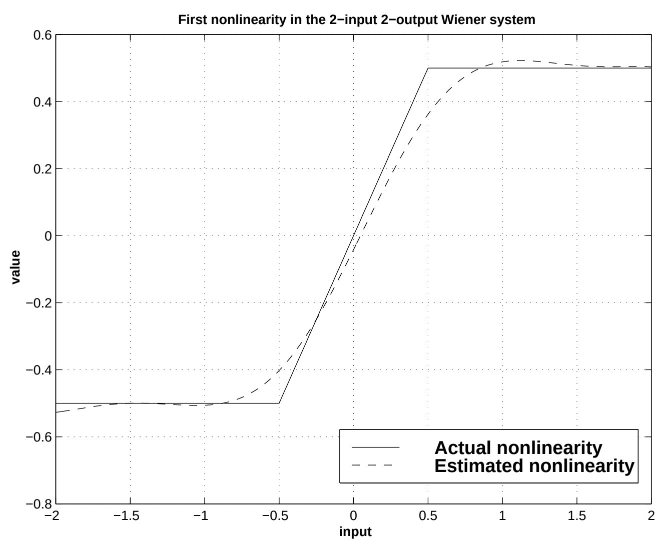

Fig. 7. First nonlinearity in the 2-input 2-output Wiener system

The ARMA models estimated the linear dynamics of the system. The values of the parameters a_1 and a_2 were converged to 0.6199 and -0.7512, respectively, which are very close to the actual ones. The true values for the zeros b_{011}, b_{012}, b_{021} and b_{022} for each of the transfer function were 0.81, 0.62, -0.53 and -0.22 that converged to 0.5333, 0.5306, -0.3568 and -0.1842, respectively. These values differ from the true ones, as these values are merely gains and are adjusted in the nonlinearities themselves. This phenomenon is well described in [33]. The plots include the compensation of these gains. The mean square error is minimized to 0.25 after around 80

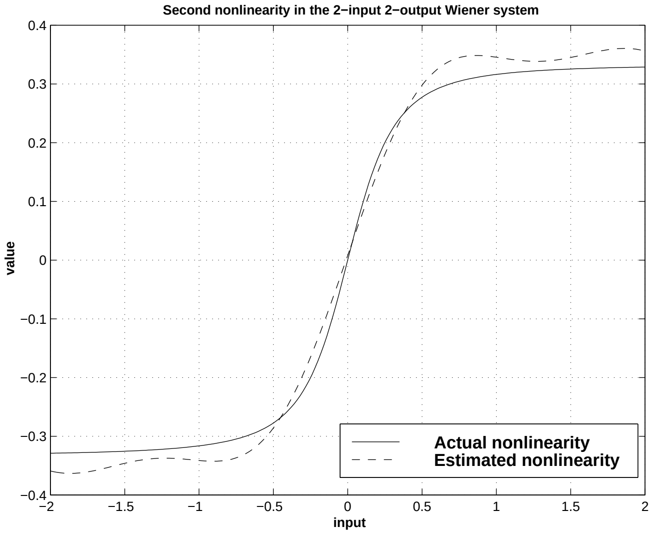

Fig. 8. Second nonlinearity in the 2-input 2-output Wiener system

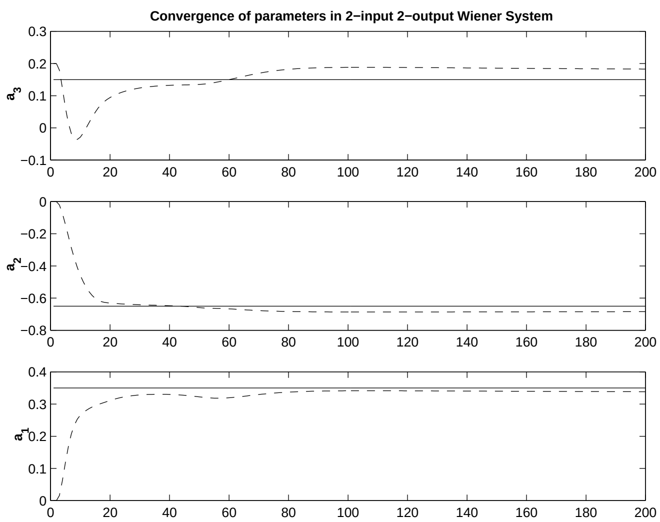

Fig. 9. Convergence of poles of the linear part in 2-input 2-output Wiener system

iterations. The mean square error plot is shown in Fig. 10. The identified static nonlinearities and truly identified poles of the linear parts reveal the capability of the identification algorithm to be applied on MIMO systems.

VI. Conclusions

A new method for identification of MIMO Wiener systems has been developed. The algorithm is an extension to the SISO Wiener model identification scheme presented in [1]. The nonlinearities are modeled using the RBFNN and the linear dynamics are modeled by MIMO ARMA model. The training algorithm is based on the LMS principle that updates the weights of RBFNN and coefficients of ARMA simultaneously. The proposed identification algorithm is applied to an example and the results show promising identification behavior.

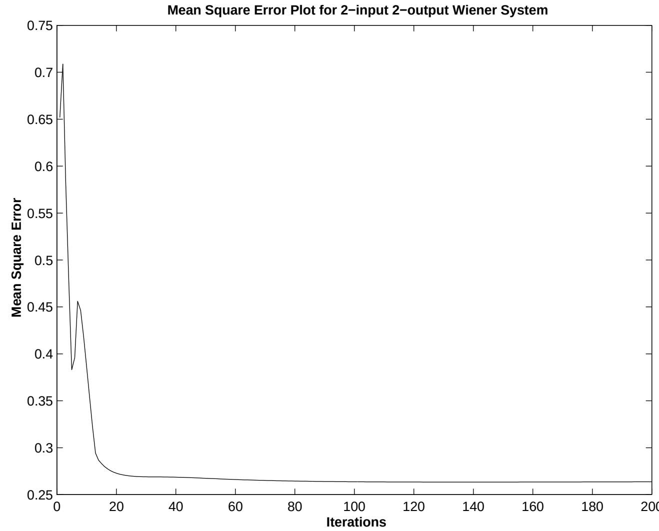

Fig. 10. Square error plot for the 2-input 2-output Wiener system

References

[1] Ali, S. Saad Azhar and Hussain N. Al-Duwaish, *Identification of Wiener Model Using Radial Basis Functions Neural Networks*, Lecture Notes in Computer Science, Vol. 24, pp 344-350, 2002

[2] Haddad, A. H. and Thomas, J. B., *Optimal and suboptimal nonlinear filters for the discrete inputs.*, IEEE transactions on Information Theory, Vol. 35, pp 409-418, 1968

[3] DeBoer, E., *Cross-correlation function of bandpass nonlinear networks*, IEEE, Vol. 16, pp 1443-1446, 1976

[4] H. Al-Duwaish and M. Nazmul Karim and V. Chandrasekar, *Hammerstein Model identification by multilayer feedforward neural networks*, International Journal of Systems Science, Vol. 28, pp 49-54, 1997.

[5] Miller, J. H. and Thomas, J. B., *Detectors for the discrete time signalsin non-Gaussian noise.*, IEEE Transactions on Information Theory, Vol. 18, pp 241-250, 1972.

[6] McCannon, T. E. and Gallagher, N. C. and Mindo-Hamedani, D. and Wise, G.L., *On the design of the nonlinear discrete time predictors.*, IEEE Transactions on Information Theory, Vol. 28, pp 366-371, 1982.

[7] Maqusi, M., *Performance of baseband signals didgital transmission in nonlinear channel with memory*, IEEE Transactions on Communications, Vol. 33, pp 715-718, 1985.

[8] Kung, M. C. and Womack, B. F., *Discrete time adaptive control of linear dynamical systems with a two-segment piece-wise-linear symmetric nonlinearity*, IEEE Transaction on Automatic Control, Vol. 29, pp 173-180, 1984.

[9] Eskinat, E. and Johnson, S. H. and Luyben, W. L., *Use of Hammerstein Models in identification of nonlinear systems*, American Institute of Chemical Engineering Journal, Vol. 37, pp 255-268, 1991.

[10] Hunter, I. W. and Korenberg, M. J., *Identification of nonlinear biological systems: Weiner and Hammerstein cascade models*, Biological Cybernetics, Vol. 55, pp 135-144, 1986.

[11] S. A. Billings and S. Y. Fakhouri, *Identification of nonlinear systems using the Wiener model*, Electronic Letters, Vol. 13, pp 502-504, 1978.

[12] D. W. Hu and Z. Z. Wang, *An identifiaction method for the wieer model of nonlinear systems*, Proceedings of the 36^{th} conference on Decision and Control IEEE, Vol. 1, pp 783-787, December 1991.

[13] H. Al-Duwaish and M. Nazmul Karim and V. Chandrasekar, *Use of Multilayer Feedforward Neural Networks in Identification and Control of Wiener Model*, IEE Proceedings: Control Theory and Applications, 3, Vol. 143, pp 225-258, 1995.

[14] Marco Lovera and miche Verhaegen, *Recursive subspace identification of linear and nonlinear type Wiener type models*, Proceedings of international conference on Control Applications IEEE, Vol. 1, pp 1250-1254, September 1998.

[15] C. T. Chou and M. Verhaegen , *Identification of Wiener Model with data pre-filtering*, Proceedings of the 38^{th} conference on Decision and Control IEEE, Vol. 1, pp 604-605, December 1999.

[16] Anna Hagenblad and Lennart Ljung, *Maximum likelihood estimation of Wiener models* IEEE Proceedings of the 39th Conference on Decision & Control, Vol. 3, pp 2417-2418, 2000

[17] K. Rodriguez-Vázquez and P. J. Fleming, *Multi-objective genetic programming for nonlinear system identification* Electronic Letters IEEE, Vol. 34, pp 930-931, April 1998.

[18] Greblicki, W., *Nonparametric Appraoch to Wiener systems identification* IEEE Transactions on Information Theory, Vol. 44, pp 538-545, 1997.

[19] Greblicki, W. and Pawlak, M., *Nonparametric identification of Hammerstein systems* IEEE Transactions on Information Theory, Vol. 35, pp 409-418, 1989.

[20] Wlodzmierz Greblicki, *Continuous-time Hammerstein system identification* IEEE transactions on automatic control, Vol. 45, pp 1232-1236, 2000.

[21] Wlodzmierz Greblicki, *Continuous-time Wiener system identification* IEEE transactions on automatic control, Vol. 43, pp 1488-1493, 1998.

[22] Anna Hagenblad, *Identification of Wiener models* First Conference on Computer Science and Systems Engineering in Linkping ECSEL, Vol. 1, pp 175-183, Jan 1998.

[23] C. T. Chou and M. Verhaegen , *Identification of Wiener Model with process noise* Proceedings of the 38^{th} conference on Decision and Control IEEE, Vol. 1, pp 598-603, December 1999.

[24] Yong Fang and T. W. S. Chow, *Orthogonal Wavelet Neural Network Applying to Identification of Wiener Model* IEEE transactions Circuit and Systems, 4, Vol. 47, pp 591-593, 2000.

[25] Patrick Celka and Neil J. Bernard and Jean-Marc Vesin, *stochastic gradient identification of Polymial Wiener systems: Analysis and Application* IEEE transactions on signal processing, Vol. 49, pp 301-312, 2001.

[26] Anders E. Nordsj and Lars H. Zetterberg, *Identification of certain time-varying Wiener and Hammerstein systems* IEEE transactions on signal processing, Vol. 49, pp 577-592, 2001.

[27] Seth L. Lacy and R. Scott Erwin Dennis S. Bernstein, *Identification of Wiener Systems with known nonivertible nonlinearities* Proceedings of the American Control Conference ACC, Vol. 1, pp 4890-4895, June 2001.

[28] Haykin, S., *Neural Networks:A Comprehensive Foundation II* Macmillan/IEEE Press, 1994, 1999.

[29] Jun-Dong, P. Jones, W. L., *Comparison of neural network classifiers for NSCAT sea ice flag* IEEE International Geoscience and remote sensing symposium proceedings., Vol. 4, pp 2237-2239, December 1998.

[30] Fortuna, L. and Muscato, G. and Xibilia, M. G., *A comparison between HMLP and HRBF for attitude control* IEEE transactions on neural networks, Vol. 122, pp 318-328, March 2001.

[31] Finan, R. A. and Sapeluk, A. T. and Damper, R. I., *Comparison of multilayer and radial basis functions neural networks for the text dependent speaker recognition* IEEE International conference on Neural Networks, Vol. 4, pp 1992-1997, December 1996.

[32] Hussain N. Al-Duwasih and Ali, S. Saad Azhar, *Identification of Hammerstein model using Radial Basis Functions Neural Networks* International Conference on Artificail Neural Networks ICANN, Vol. 1, pp 951-956, August 2001.

[33] S. Saad Azhar Ali, *Block Oriented Identification: A neural network approach towards identification of Hammerstein and Wiener Models* Lambert Academic Publishing, 2010.