Performance Analysis of Series Expansions for Nonlinear SISO System Identification using RLS Algorithm

D. Yugandhar¹ and S.K. Nayak²

¹ Aditya Institute of Technology and Management, E.C.E Department, Tekkali, Andhra Pradesh, India
Email: yugndasari@gmail.com
² Berhampur University, Electronic Science Department, Berhampur, Odisha, India
Email: sknayakbu@rediffmail.com

Abstract—This paper presents the modified bilinear volterra adaptive filter technique without considering the cross correlated input and output terms for nonlinear system identification. The result obtained by our proposed technique is compared with trigonometric, power and projected wavelet series expansions. In the present implementation the identification problem is performed on four series expansions using bilinear RLS algorithm for training the weights of recurrent FLANN structure. For simplicity and to avoid instability in the convergence, this model provides better mean square error convergence and less computational complexity even though the training samples used contain outliers.

Index Terms—Nonlinear system identification, outliers, recurrent functional link artificial neural network, volterra filter, bilinear filter, subband filtering, bilinear RLS algorithm.

I. INTRODUCTION

In system identification, the principle relies on the estimation of transfer function of an unknown system based on measurement of error signal which is to be minimized using different adaptive algorithms. System identification plays an important role in biological systems, telecommunication systems, control systems, instrumentation, signal and image processing. Many polynomial nonlinear systems obey the principle of superposition but not that of homogeneity. It is also true that for linear time invariant systems (LTI), the frequency component present in the output signal are the same as those present in the input signal. However, for nonlinear systems [8], the frequency in the output signal are not typically the same as those present in the input signal but also contains other new frequencies. The traditional Least Mean Square algorithm is well suited for identification of linear static systems. However, in practice most of the systems are nonlinear and dynamic. The conventional linear approaches do not achieve very satisfying performances which requires the development of nonlinear approaches and advanced methods. The non recursive polynomial model, based on the volterra series expansion [9] and recursive polynomial approaches based on adaptive bilinear filtering has been used for nonlinear system identification problem. Many neural network [1] concepts have been applied to deal with nonlinear system identification problems such as multilayer perceptron algorithm (MLP) [3],

© Elsevier, 2014
radial basis function algorithm (RBF) [5], and complex radial basis Function algorithm [2]. To improve the identification performance of nonlinear systems various evolutionary algorithms (EA) [10] have been reported. Unlike other optimization techniques, the EAs are a population-based search algorithms [11] which work with a population of chromosomes or particles that represent different potential solutions. The identification problem also investigated using nonlinear dynamic series-parallel model using differential evolution (DE) [12] for training the weights of FLANN [4] model.

In this paper, we demonstrated two new series expansions to identify a given nonlinear system in the presence of outliers in the training samples. The rest of the paper is organized as follows. In Section II, we introduced an adaptive identification scheme for updating the model parameters, detail explanation of series expansions like trigonometric, power series including the two proposed series expansions and also algorithm implementation. In section III, we presented SISO identification problem and simulation results. Finally, Section IV provided the concluding remarks.

II. PROPOSED ADAPTIVE IDENTIFICATION SCHEMES

The block diagram of an adaptive nonlinear system identification scheme is shown in Fig. 1. Let any time instant $n$, $x(n)$, $y(n)$ and $\hat{y}(n)$ be the input, output of the plant and estimated output of the model respectively. The difference of these two outputs produces an error $e(n)$. This figure also reflects the nonlinear model, which can be analysed by the amount of MSE and computational cost of the different series expansions. To investigate the problem of SISO nonlinear system identification in (13) & (14), two input samples $x(n), x(n-1)$ and three output samples $y(n), y(n-1), y(n-2)$ are considered and represented in the following manner for simplicity.

$$X(n) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x(n) & x(n-1) \end{bmatrix}^T$$

$$Y(n) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} y(n) & y(n-1) & y(n-2) \end{bmatrix}^T$$

$$XY(n) = \begin{bmatrix} X(n) \\ Y(n) \end{bmatrix}^T = \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}^T$$

In this paper we considered input $[x_1, x_2, y_1, y_2, y_3]^T$ to the recurrent FLANN structure shown in Fig. 2. The estimated output of the model at $n+1$ time instance is $\hat{y}(n+1) = \phi(XY(n))W(n)$ where $\phi(.)$ is a function of all variables mentioned above and $\phi(XY(n))$ is nonlinear expansion of input $XY(n)$ and $W(n) = [w_{x1} w_{x2} w_{y1} w_{y2} w_{y3}]^T$ is the weight vector at $n$th time instance of the estimated model. These weights are updated based on error $e(n+1)$ obtained from the model using bilinear RLS algorithm[8].

![Figure 1. Adaptive system identification model using RLS algorithm](image)

The input to the functional expansion block expanded by trigonometric series with 26 inputs and power series with 16 inputs as given in (1) and (2) respectively.
\[ \phi(XY(n)) = \\
\begin{bmatrix}
\phi(x_1) & \phi(x_2) & \phi(y_1) & \phi(y_2) & \phi(y_3)
\end{bmatrix}^T
\]
\[= [1 \ x_1 \ x_2 \ y_1 \ y_2 \ y_3 \ \sin(\pi x_1) \ 
\cos(\pi x_1) \ \sin(\pi x_2) \ \cos(\pi x_2) \ \sin(\pi y_1) \ 
\cos(\pi y_1) \ \sin(\pi y_2) \ \sin(\pi y_2) \ \cos(\pi y_2) \ 
\sin(\pi y_3) \ \cos(\pi y_3) \ \sin(2\pi x_1) \ 
\cos(2\pi x_1) \ \sin(2\pi x_2) \ \cos(2\pi x_2) \ 
\sin(2\pi y_1) \ \cos(2\pi y_1) \ \sin(2\pi y_2) \ 
\cos(2\pi y_2) \ \sin(2\pi y_3) \ \cos(2\pi y_3) \]^T \]

We proposed two new set of training series namely projected wavelet series expansion and modified bilinear volterra series expansions and the concept is presented below.

A. Wavelet Projections

In this section, a brief review of the wavelet theory [6] is given.

S2.1): Any signal \(x(t)\) in continuous time domain can be represented in terms of wavelet basis function as

\[ x(t) = \sum_{k \in \mathbb{Z}} c_{(0k)j} \phi_{0k}(t) + \sum_{j=0}^{\infty} \sum_{k \in \mathbb{Z}} d_{jk} \psi_{jk}(t) \]  \hspace{1cm} (3)

Here \(\mathbb{Z}\) represents the set of integers and \(j\) and \(k\) are the scaling and translation parameters respectively.

The \(c_{0k}\)'s are normally called approximation or scaling coefficients and \(d_{jk}\)'s are referred to as the detail or wavelet coefficients.

S2.2): \(\{\phi_{jk}(t) = 2^{j/2} \phi(2^j t - k)\}_{k \in \mathbb{Z}}\) is an orthonormal basis derived from the scaling function \(\phi(t)\) for the subspace \(V_j \subset V_{j+1}\).

S2.3): \(\{\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)\}_{k \in \mathbb{Z}}\) is a wavelet function constitute an orthonormal basis for the subspace \(W_j = V_{j+1} - V_j\).

![Figure 2: Structure of recurrent FLANN model](image-url)
S2.4): These subspaces define a multiresolution analysis on $L^2(R)$ if the following additional properties are to be satisfied like, $\bigcap V_j = \{0\}, \bigcup V_j$ is dense in $L^2(R)$.

S2.5): $s(t) \in V_j \Leftrightarrow s(2^j t) \in V_{j+1}$ for all $j \in \mathbb{Z}$. S2.6): We can construct the scaling function coefficients $(c_{jk})$ and wavelet coefficients $(d_{jk})$ by using the relations which are expressed as follows

$$c_{jk} = \int x(t) \varphi_{jk}(t) dt = \langle x(t) \varphi_{jk}(t) \rangle$$  \hspace{1cm} (4)

$$d_{jk} = \int x(t) \psi_{jk}(t) dt = \langle x(t) \psi_{jk}(t) \rangle$$  \hspace{1cm} (5)

S2.7): The discrete wavelet coefficients of the function $f(x)$ obtained by

$$W_{\varphi}(j_0,k) = \frac{1}{\sqrt{M}} \sum_x f(x) \varphi_{j_0,k}(x)$$  \hspace{1cm} (6)

$$W_{\psi}(j,k) = \frac{1}{\sqrt{M}} \sum_x f(x) \psi_{j,k}(x)$$  \hspace{1cm} (7)

Figure 3: Daubechies 4 scaling function

Figure 4: Daubechies 4 wavelet function
S2.8): The function can be constructed for \( j \geq j_0 \)

\[
f(x) = \frac{1}{\sqrt{M}} \sum_{k} W_\varphi(j0,k) \varphi_{j0,k}
\]

\[
+ \frac{1}{\sqrt{M}} \sum_{j=j0}^{\infty} \sum_{k} W_\psi(j,k) \psi_{j,k}(x)
\]

(8)

Here, \( f(x), \varphi_{j0,k}(x), \psi_{j,k}(x) \) are functions of the discrete variables \( x = 0, 1, 2, 3, \ldots, M - 1 \).

S2.9): There are many types of wavelets, we have chosen an orthogonal wavelet \( db4 \) is having properties compact support, robust, fast, adaptable and a useful basis function for the projected wavelet decomposition approach and its scaling and wavelet functions are shown in Figs. 3-4 respectively. The two level decomposition of the input sequence \( X(n) = [x_1 \ x_2]^T \) is obtained using \( db4 \) wavelet and the decomposition procedure is explained below.

\[
X(n) = \frac{1}{2} [W_\varphi(0,0) \varphi_{0,0}(n) + W_\psi(0,0) \psi_{0,0}(n) + W_\varphi(1,0) \varphi_{1,0}(n) + W_\psi(1,1) \psi_{1,1}(n)]
\]

and

\[
a_2 X(n) = \frac{1}{2} [W_\varphi(0,0) \varphi_{0,0}(n)] \quad d_2 X(n) = \frac{1}{2} [W_\psi(0,0) \psi_{0,0}(n)]
\]

\[
d_1 X(n) = \frac{1}{2} [W_\varphi(1,0) \varphi_{1,0}(n) + W_\psi(1,1) \psi_{1,1}(n)]
\]

\( a_2 X(n), d_2 X(n) \) and \( d_1 X(n) \) are known as level 2 approximate coefficients, level 2 detail coefficients and level 1 detail coefficients of input sequence \( X(n) \) respectively.

\[
X(n) = [a_2 X(n) \ d_2 X(n) \ d_1 X(n)]
\]

\[
= [a_2 x_1 a_2 x_2 d_2 x_1 d_2 x_2 d_1 x_1 d_1 x_2]^T
\]

\( d_2 X(n) = [d_2 x_1 d_2 x_2]^T \)

S2.10): Similarly, two level decomposition of the output sequence \( Y(n) = [y_1 \ y_2 \ y_3]^T \) using \( db4 \) wavelet is given below

\[
[a_2 Y(n) \ d_2 Y(n) \ d_1 Y(n)] = [a_2 y_1 \ a_2 y_2 \ a_2 y_3, d_2 y_1, d_2 y_2, d_2 y_3, d_1 y_1, d_1 y_2, d_1 y_3]^T
\]

\( a_2 Y(n), d_2 Y(n) \) and \( d_1 Y(n) \) are known as level 2 approximate coefficients, level 2 detail coefficients and level 1 detail coefficients of the output sequence \( Y(n) \) respectively.

\[
a_2 Y(n) = [a_2 y_1 \ a_2 y_2 \ a_2 y_3]^T
\]

\[
d_2 Y(n) = [d_2 y_1 \ d_2 y_2 \ d_2 y_3]^T
\]

\( d_1 Y(n) = [d_1 y_1 \ d_1 y_2 \ d_1 y_3]^T
\)

S2.11): Therefore, the expanded sequence of the projected wavelet series having a total of 21 input samples are shown below

623
\[\phi(XY(n)) = \\
[1 \ x_1 \ y_1 \ y_2 \ y_3 \ a_2 x_2^2 \ a_2 y_2^2 \ a_3 x_3^2 \ a_3 y_3^2 \\
\ a_2 x_2 y_2^2 \ d_2 x_2^2 \ d_2 y_2^2 \ d_2 x_2^3 \ d_2 y_2^3 \ d_1 x_1^2 \ d_1 x_1^3, \\
\ d_1 y_1^2 \ d_1 y_2^2 \ d_1 y_3^2]^T \]

By considering the input vector in (9) the weights of the model are updated using bilinear RLS algorithm and
the results are compared.

S2.12): An ECG signal is of 714 samples decomposed using above procedure. Level 2 approximations, level
2 details and level 1 detail are plotted in Fig. 5. From the figure we can conclude that wavelet decomposition
divides the signal into different frequencies and the training process will be more accurate because it can
adapt changes to all the frequency content present in the signal easily.

Figure 5: A wavelet series expansion of a given ECG Signal

B. Modified Bilinear Volterra Adaptive Filters

Volterra Filters

The polynomial models of nonlinear systems known as volterra series model have become quite popular in
adaptive nonlinear filtering methods in the last few years. The volterra series representation is an extension of
linear system theory with high complex nature of nonlinear filtering. The truncated \(P\)th order volterra series
expansion is shown below.

\[y(n) = \\
\ h_0 + \sum_{m_1=0}^{N-1} h_1[m_1 x[n-m_1] \\
+ \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} h_2[m_1, m_2 x[n-m_1] x[n-m_2] \\
+ \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \sum_{m_3=0}^{N-1} h_3[m_1, m_2, m_3 x[n-m_1] x[n-m_2] x[n-m_3] \\
+ \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \sum_{m_3=0}^{N-1} \sum_{m_p=0}^{N-1} h_p[m_1, m_2, \ldots, m_p x[n-m_1] x[n-m_2] \ldots x[n-m_p] \ldots x[n-m_p]] \quad (10) \]
where $h_k(), 0 \leq k \leq p$ represents kernel coefficients, $x(.)$ and $y(.)$ represents volterra input and output signals respectively. From (2), the volterra series expansion can be interpreted as a Taylor series expansion with memory. For $h_0 = 0$ and order $p = 2$ the coefficient vector $H[n]$ and signal $X(n)$ input vectors are defined as.

$$H[n] = h_0[0; n]\ldots h_2[0, N; n],$$

$$X[n] = [x[n], x[n-1], \ldots x[n-N+1], x^2[n],$$

$$x[n], x[n-1], \ldots x[n-N+1], x^2[n-1], \ldots$$

where $N, M, I$ and $L$ are the orders of the adaptive filter. The coefficient vector can be described as.

$$[h_0[0; n], h_1[1; n], \ldots h_k[N-1; n], h_2[0, 0; n],$$

$$y(n) = \sum_{m=0}^{M} b_m(n)x(n-m) - \sum_{j=1}^{N} a_j(n)y(n-j) + \sum_{i=0}^{J} \sum_{j=1}^{L} c_{ij}(n)x(n-i)y(n-j)$$

where $b_m(n)$, $c_{ij}(n)$, $a_j(n)$ are the coefficient of the adaptive bilinear filter at time instance $n$. The signal input vector is defined by

$$X(n) = [x(n), x(n-1), \ldots x(n-M), y(n-1), y(n-2)$$

$$\ldots y(n-N), x(n)y(n-1), \ldots$$

$$x(n-I)y(n-L+1), x(n-I)y(n-L)]$$

Modified Bilinear Volterra Filters

Equations (10) and (11) modified without considering the cross correlated input and output terms is given in (12) is called as modified bilinear volterra filter series expanded to 15 inputs.

$$\phi(XY(n)) = [1, x_1, x_2, y_1, y_2, y_3, x_1^2, x_2^2, y_1^2,$$

$$y_1y_2, y_2y_3, y_3y_1, y_2^2, y_3^2]$$

(12)
C. Algorithm Implementation

0% of outliers in the training samples

Step 1: In this present investigation a series parallel identification model is used with system input order of 2 and output order 3 are considered for the experiment.

Step 2: A uniformly distributed random signal of 800 samples is generated over the interval of [-1 1] applied to the actual nonlinear dynamic plant and the desired plant output is obtained by (13) and (14).

Step 3: The samples applied to the plant also applied to the adaptive model simultaneously. The input and output training samples are expanded using the trigonometric series given in (1) and the plant input vector is considered as \( \phi(XY(n)) \) and weight vector of length \( M = 26 \) initialized with zeros before the training begins.

\[ w(0) = 0 \]

Step 4: \( \delta \) is regularization parameter initialised with the values between 0 to 100, \( \lambda \) is forgetting factor initialised with 0.999 and \( P(0) = \delta^{-1}I \) is obtained where I is identity matrix of size \( M \times M \).

Step 5: For each instant of time \( n \), the following are calculated [7]

\[ \pi(n) = P(n-1)X(n) \]

\[ k(n) = \frac{\pi(n)}{\lambda + X^H(n)\pi(n)} \]

\( k(n), P(n) \) are known as Kalman gain and inverse correlation matrix respectively.

Step 6: An error \( e(n) \) is obtained by subtracting desired output and estimate output of the plant, \( H \) indicates the Hermitian operator.

\[ e(n) = y(n) - w^H(n-1)X(n) \]

Step 7: Weights vector and inverse correlation matrix values are updated using the following equations

\[ w(n) = w(n-1) + k(n)e^*(n) \]

\[ P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)X^HP(n-1) \]

Step 8: The updation process in the step 5 to 7 continued till it covers all the 800 input samples.

Step 9: Save the obtained weight vector to test the system.

After training phase is over, the effectiveness of the proposed model is evaluated by using the test signal which is given in (16) with the weights obtained during the training process. The above procedure is repeated for power series using inputs 16 according to (2) projected wavelet series using inputs 21 according to (9) and modifies bilinear volterra series using 15 coefficients according to (12). The performance of all the four series expansions using bilinear RLS algorithm is evaluated by finding mean squared error using (17).

10% of outliers in the training samples

The above training and evaluation procedure is repeated for adding 10% of the training samples over the interval of [-1 1] is added to the actual plant output in step 2 and the weights are obtained for all the four series expansion using bilinear RLS algorithm.

40% of outliers in the training samples

The above training and evaluation procedure is repeated for adding 40% of the training samples over the interval of [-1 1] is added to the actual plant output in step 2 and the weights are obtained for all the four series expansion using bilinear RLS algorithm. The nonlinear system identified by adding 40% outliers in the training samples for trigonometric, power, projected wavelet and bilinear series are given in (18) - (21) and the plant output and model output are shown in Figs. 6-9 respectively.

III. Simulation And Results

For nonlinear dynamic plant described by difference equation in (13) & (14), extensive simulation studies have been carried out. In this investigation a series –parallel model in (15) is used along with bilinear RLS
algorithm for training weights of FLANN structure. The performance of the proposed modified bilinear
volterra model is then compared with trigonometric, power and projected wavelet series. In this model an
uniformly distributed random signal of 800 sample over the interval [-1, 1] is used as input for the training
process and the result converges for single iteration.
\[
y(n + 1) = f[y(n), y(n - 1), y(n - 2), x(n), x(n - 1)]
\] (13)

where the unknown non linear function \( f \) is given by
\[
f[a_1, a_2, a_3, a_4, a_5] = \frac{a_1 a_2 a_3 (a_1 - 1.0) + a_4}{1.0 + a_2^2 + a_3^2}
\] (14)
\[
y(n + 1) = \phi[y(n), y(n - 1), y(n - 2), x(n), x(n - 1)]
\]
(15)

where \( \phi(.) \) is a function of all variables mentioned above. The testing is analysed by parallel scheme. The
input to the identified model is given as
\[
x(n) = \begin{cases} 
\sin \frac{2\pi n}{250} & \text{for } n \leq 500 \\
0.8 \sin \frac{2\pi n}{250} + 0.2 \sin \frac{2\pi n}{25} & \text{for } n > 500 
\end{cases}
\] (16)

Mean squared error measurement shown below is used for comparison of all the four series expansions.
\[
MSE = \sum_{i=1}^{N} e(i)^2, \text{ where } N = 800
\]
(17)

The bilinear RLS algorithm is implemented for a given two input and three outputs SISO dynamic system
which is given in (13) & (14). The difference equation for trigonometric series with 26 inputs is given below
\[
y(n + 1) = 0.008222 - 0.07565x_1 \\
+ 0.761585x_2 + 0.007206y_1 \\
- 0.00585y_2 + 0.019337y_3 \\
+ 0.019501\sin(\pi x_1) - 0.04233\cos(\pi x_1) \\
+ 0.104216\sin(\pi x_2) - 0.06374\cos(\pi x_2) \\
+ 0.009398\sin(\pi y_1) + 0.037366\cos(\pi y_1) \\
- 0.06609\sin(\pi y_2) - 0.07158\cos(\pi y_2) \\
+ 0.107566\sin(\pi y_3) + 0.004275\cos(\pi y_3) \\
+ 0.010181\sin(2\pi x_1) - 0.03929\cos(2\pi x_1) \\
+ 0.009435\sin(2\pi x_2) + 0.006953\cos(2\pi x_2) \\
- 0.02347\sin(2\pi y_1) - 0.00925\cos(2\pi y_1) \\
+ 0.016458\sin(2\pi y_2) + 0.013668\cos(2\pi y_2) \\
- 0.00908\sin(2\pi y_3) - 0.01979\cos(2\pi y_3)
\]
The difference equation obtained using power series with 16 inputs and projected wavelet series with 21 inputs are given in (19) and (20) respectively. We can conclude by analysing Figs. 6-8, the adaption process is better for projected wavelet series even though the computational cost is more comparatively trigonometric and power series expansions.

\[
y(n + 1) = 0.007144 - 0.07235x_1 \\
+ 0.746556x_2 - 0.06399y_1 \\
+ 0.031513y_2 + 0.072502y_3 \\
- 0.00306x_1^2 + 0.007678x_2^2 \\
- 0.01428y_1^2 - 0.01412y_2^2 \\
- 0.01062y_3^2 + 0.036479x_1^4 \\
- 0.02714x_3^3 + 0.090016y_1^3 \\
- 0.08078y_2^3 - 0.00525y_3^3
\]

(19)

\[
y(n + 1) = \\
- 0.0043 - 0.0705x_1 + 0.7586x_2 \\
- 0.0278y_1 + 0.082y_2 + 0.043y_3 \\
- 0.2429x_2x_3^2 - 0.1303x_1y_1^2 - 0.1028x_2y_2^2 \\
+ 0.0692x_2y_3 - 0.0130f_1x_1^2 + 0.1227f_2x_2^2 \\
+ 0.0833f_3y_2^2 + 0.0025f_4y_3^2 + 0.0159f_5y_5^2 \\
- 0.0204f_6x_1^2 + 0.0085f_7x_2^2 - 0.0603f_8y_1^2 \\
+ 0.1526f_9y_2^2 + 0.1136f_10y_3^2
\]

(20)

The difference equation obtained using modified bilinear volterra with 15 inputs is given below

\[
y(n + 1) = \\
0.0078 - 0.0717x_1 + 0.7527x_2 \\
- 0.0371y_1 + 0.0076y_2 + 0.0429y_3 \\
- 0.0075x_1^2 - 0.0140x_1x_2 \\
- 0.0400x_2^2 - 0.0826y_1^2 \\
- 0.1510y_1y_2 + 0.0255y_2y_3 \\
- 0.1086y_1y_3 + 0.1422y_2^2 \\
- 0.0004y_3^2
\]

(21)
Figure 7: Comparison of response of the dynamic plant using power series expansion

Figure 8: Comparison of response of the dynamic plant using modified wavelet series expansion

Table I compares mean squared error performance using (17) for all series expansions namely trigonometric, power, projected wavelet and modified bilinear volterra series expansions using bilinear RLS algorithm by adding 0%, 10% and 40% outliers in the training samples.

Figure 9: Comparison of response of the dynamic plant using modified bilinear series expansion
TABLE I. COMPARISON OF MEAN SQUARED ERROR FOR ALL THE FOUR SERIES EXPANSIONS BY ADDING DIFFERENT PERCENTAGE OF OUTLIERS IN THE TRAINING SAMPLES

<table>
<thead>
<tr>
<th>S.NO</th>
<th>Percentage of Outliers</th>
<th>Mean Squared Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Trigonometric Series Expansion</td>
</tr>
<tr>
<td>1.</td>
<td>0</td>
<td>&gt; 0.0050</td>
</tr>
<tr>
<td>2.</td>
<td>10</td>
<td>&gt; 0.0050</td>
</tr>
<tr>
<td>3.</td>
<td>40</td>
<td>&gt; 0.0050</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

The adaptive nonlinear system identification task by transforming input samples using all the four series expansion is carried. Out of all the four series expansions, modified bilinear volterra series expansion gives good results, fast convergence and less computational complexity even in the presence of outliers in the training samples. This can be observed from the table that MSE is constant even though percentage of outliers in the training samples increased to 10% and 40%. The Projected wavelet series also gives competitive results comparatively traditional series like trigonometric, power series expansions because of their nonlinear behaviour of decomposition. The mean squared error can be reduced further by increasing the computational complexity by introducing hidden layer in between input and the output of the model.

REFERENCES