Adaptive Volterra Modeling For Nonlinear Systems Based on LMS Variants

Sachin N. Kapgate*,1, Saurav Gupta *,2, Ajit Kumar Sahoo*,3
*Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela, India
Email: [1]sachin.kapgate123@gmail.com, [2]greater.saurav@gmail.com, [3]ajitsahoo@nitrkl.ac.in

***Abstract*—Many practical systems tend to have some extent of nonlinearity involved in their behavior. System identification and control design for nonlinear dynamical systems is achieving extensive attention in many practical applications. To model nonlinear system behavior, many mathematical models have been developed and employed in practical applications. This paper mainly focuses on application of least mean square (LMS) variants for the adaptive implementation of one such model known as nonlinear Volterra model. The challenging problems involved with the stability and convergence rate of traditional least mean square based approach are shown individually. The supporting analysis and simulations are provided to justify the efficacy of presented work. Least mean square variants based Volterra modeling approaches can be effectively applied in system control design, acoustic echo cancellation and stability analysis of the nonlinear systems.**

***Index Terms*—Signal processing, nonlinear System modeling, Volterra model, Least mean squares.**

I. Introduction

System modeling can be delineated as the process of constructing mathematical models of the system using limited number of input-output measurement data of the system. These models are important elements for control system designs, stability assessments and automation through signal processing. In practice, most of the systems we encounter typically contain at least some extent of nonlinearity in their dynamics [1]. In many modeling methodologies, nonlinearity in the behavior of system has been approximated by a linear relation in order to avoid the underlying complexity in trade-off with modeling accuracy [2]. Since in the control and automation fields, the accuracy of modeling is vital for the reliable performance of the system, more accurate nonlinear models are gaining importance in the relevant fields.

The major classification of nonlinear system modeling approaches leads to parametric and non-parametric models [3]. Parametric models as discussed in [4], need some assumptions or some porior knowledge of the system behavior and are known to be white-box and grey-box models accordingly. Unlike parametric models, nonparametric models do not require prior knowledge about system and use only the input-output observation data of the system. The nonparametric models may not be able to model highly complex systems efficiently as obtaining knowledge about the system may not be possible in many practical cases. Hence, the input-output observation data based models are often required for modeling and control of such systems. The Volterra and Wiener series based nonparametric models are widely employed in practical applications such as acoustic echo cancellation [5], rotor-bearing systems, controller designs [6] etc. In this work, Volterra model is considered for nonlinear system modeling.

The Volterra model comprises of a series of higher-order kernels which can be interpreted as higher-order approximations of the system's impulse response. The order of traditional Volterra model is usually considered to be infinite. But due to infinite order, the model is highly complex and impractical. In many practical applications, the finite-order approximation of Volterra model is sufficient to model the nonlinear dynamical systems [7], [8]. The truncated finite-order approximation of the Volterra model makes it practicable and has less complexity involved in the implementation and understanding [9]. Many practical systems tend to have fading memory characteristics in their dynamics. This means that the response of system is influenced by the present input and only few of the past inputs [10]. This helps to further reduce the complexity of Volterra series based system modeling.

For effective employment in real-time applications, the models need to be adaptive in nature [11]. The neural network based Volterra modeling approach is adaptive in nature but has very high complexity involved in the modeling [12]. In this article, least mean square (LMS) variants based Volterra modeling approach is considered to address aforementioned issues. The traditional LMS based Volterra modeling approach presented in [13], [14] has slow convergence rate. Also, the instability in the performance of LMS based Volterra model has not been addressed. The effect of learning factor on the convergence rate and stability of the LMS based Volterra model is analyzed in this article under different operating conditions. The learning factor has upper and lower bound to guarantee the stable performance of the LMS based Volterra model and if the learning factor is not within these bounds the stable operation of model can not be guaranteed [15]. Again, when the input with very large eigenspread (ratio of largest and smallest eigenvalue) than training input data is presented to the LMS algorithm, the model parameters grow unbounded [16], [17]. This is known as weight-drifting [18].

In this article, the slow convergence rate and weight-drifting issues in traditional LMS based Volterra model are shown individually and simulation results are presented for the same. To overcome the weight-drifting problem in traditional Volterra model, leaky LMS based approach is employed where a small leakage factor is introduced in the parameter update equation. To improve the convergence speed, the modified leaky LMS based Volterra model parameter estimation approach is employed.

The notations followed to represent different entities in this article are as follows: any symbol with bar head is used for representing a vector and small case alphabets represent a scalar quantity.

The article is presented in six different sections and are briefed as follows: In Section I, the existing methodologies and related issues are discussed to justify the motivation towards this work. The mathematical model of Volterra series based nonlinear system modeling approach is formulated in Section II. In Section III, LMS based Volterra model parameter estimation approach is presented. Sections IV and V present the leaky LMS based and modified leaky LMS based Volterra model parameter estimation approaches respectively. In Section VI, numerical examples are simulated to justify the effectiveness of above methodologies and simulation results has been presented. Finally, the conclusive comments are provided in Section VII.

II. FORMULATION OF VOLTERRA SERIES BASED NONLINEAR SYSTEM MODELING

Any 1^{st}-order linear system can be modeled in terms of the input signal and the system's impulse response as given below [19].

$$d(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau \ , \tag{1}$$

where $h(t)$ is the linear impulse response of the system, $x(t)$ is the instantaneous input and $d(t)$ is the corresponding instantaneous output. And the above equation can be interpreted as usual one-dimensional linear convolution similar to that of a stable LTI system.

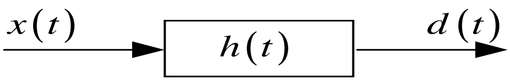

Fig. 1: LTI system model

But modeling the nonlinear systems is a much more complex problem, as usually only the linear impulse response of the system is known.

In this setup, to model the nonlinear systems, we consider the traditional Volterra model. It is implemented as a series of nonlinear volterra kernels entailing kernels from 1^{st}-order to higher-order volterra kernels. The linear impulse response $h(t)$ in (1) can be considered as 1^{st}-order Volterra kernel.

Now the nonlinear system can be modeled using traditional Volterra model as follows

$$d(t) = \mathrm{H}\{x(t)\} + \eta(t) \ , \tag{2}$$

here H is the higher-order Volterra operator and can be represented as $\mathrm{H_r} = [h_1, ..., h_r]$ and h_r represents the r^{th}-order Volterra kernel.

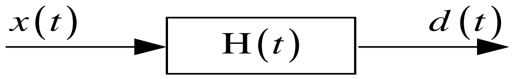

Fig. 2: Volterra model of the system

The Volterra model can be further defined as [19].

$$d(t) = \sum_{r=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_r(\tau_1, \tau_2, \ldots, \tau_r) \prod_{i=1}^{r} x(t-\tau_r)\, d\tau_r \ , \tag{3}$$

above equation represents the infinite-order Volterra model for a nonlinear continuous time-invariant system. Further, the Volterra model in (3) can be rewritten to model any causal nonlinear system as

$$d(t) = \sum_{r=1}^{\infty} \int_{0}^{\infty} \cdots \int_{0}^{\infty} h_r(\tau_1, \tau_2, \ldots, \tau_r) \prod_{i=1}^{r} x(t-\tau_r)\, d\tau_r \ , \tag{4}$$

In the above model representation i.e. (4), time can be discretized for ease of understanding and reducing the complexity in practical implementation of model. Therefore, the equation (4) can be represented in discretized form as follows [19]

$$d(t) = \sum_{r=1}^{\infty} \sum_{\tau_1=0}^{\infty} \cdots \sum_{\tau_r=0}^{\infty} h_r(\tau_1, \tau_2, \ldots, \tau_r) \prod_{i=1}^{r} x(t-\tau_r) \ . \tag{5}$$

In most of the practical systems, the system output is mostly influenced by present input and few of the past inputs. So, the model parameters and complexity can be further reduced by introducing finite fading-memory factor [10]

$$d(t) = \sum_{r=1}^{\infty} \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_r=0}^{M-1} h_r(\tau_1, \tau_2, \ldots, \tau_r) \prod_{i=1}^{r} x(t-\tau_r) \ , \tag{6}$$

where M is fading memory factor of the system, and it can be considered appropriately based on the system to be modeled.

The implementation of above infinite-order Volterra model is impractical because of the infinite number of parameters involved in the modeling and the high complexity of the Volterra kernels. The model accuracy and complexity depends on the order of Volterra kernel and hence it has to be selected as the trade-off between the former two [7], [8]. So, the infinite-order Volterra kernel has to approximated to a finite-order Volterra model which is given as follows

$$d(t) = \sum_{r=1}^{R} \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_r=0}^{M-1} h_r(\tau_1, \tau_2, \ldots, \tau_r) \prod_{i=1}^{r} x(t-\tau_r) \ , \tag{7}$$

here R is the order of the nonlinear Volterra model. As R increases the accuracy of Volterra model improves but, consequently the modeling complexity increases.

In our work, we have considered second-order approximation of Volterra model for modeling of nonlinear system. The second-order approximation of Volterra model is described below

$$\begin{aligned} d(t) = &\sum_{\tau_1=0}^{M-1} h_1(\tau_1)\, x(t-\tau_1) \\ &+ \sum_{\tau_1=0}^{M-1} \sum_{\tau_2=0}^{M-1} h_2(\tau_1, \tau_2)\, x(t-\tau_1)\, x(t-\tau_2) \ , \end{aligned} \tag{8}$$

where h_1 and h_2 are the 1^{st}-order and 2^{nd}-order Volterra kernels respectively.

The linear regression form of the model given in (8) can be written as

$$d\left(t\right) = \bar{\Theta}^T \bar{X}\left(t\right) + \eta\left(t\right) \ , \tag{9}$$

where $\bar{\Theta}$ and $\bar{X}$ are given as,

$$\Theta \ = \ \left[\bar{h}_1, \bar{h}_2\right]^T \in {}^{M+M^2} \ , \tag{10}$$

$$\bar{X}\left(t\right) = \left[\begin{array}{l} x\left(t\right), ..., \ x\left(t-\left(M-1\right)\right), \ x^2\left(t\right), ..., \\ x\left(t\right) x\left(t-\left(M-1\right)\right), ..., x^2\left(t-\left(M-1\right)\right) \end{array} \right]^T \ , \tag{11}$$

and

$$\bar{h}_1 = \left[h_1\left(1\right), ..., h_1\left(M\right)\right] \in {}^M \ , \tag{12}$$

$$\bar{h}_2 = \left[h_2\left(1\right), ..., h_2\left(M^2\right)\right] \in {}^{M^2} \ . \tag{13}$$

The estimate of optimal Volterra model parameters $\bar{\Theta}_{opt}$ can be obtained iteratively by updating $\bar{\Theta}_{est}$ at each instant using the instantaneous estimation error as follows

$$d_{est}\left(t\right) = \bar{\Theta}^T_{est}(t)\bar{X}\left(t\right) \ , \tag{14}$$

$$e\left(t\right) = d\left(t\right) - d_{est}\left(t\right) \ . \tag{15}$$

Fig. 3 gives the overview of Volterra model parameter estimation approach,

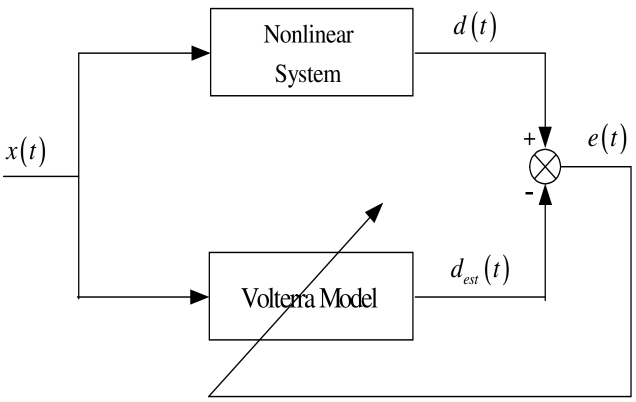

Fig. 3: Volterra model parameter estimation

Least mean square algorithm is used to update the Volterra model parameters at each instant.

III. Least Mean Square Algorithm

The cost function for LMS based Volterra model parameter estimation approach is represented as follows

$$C\left(t\right) = \sum_{t=1}^{T} \left| d\left(t\right) - \bar{\Theta}^T_{est}(t)\bar{X}\left(t\right) \right|^2 \ , \tag{16}$$

where T denotes the total number of data samples and $\bar{\Theta}_{est}(t)$ is the instantaneous Volterra model parameter estimate. To minimize the above quadratic cost function and find optimum value of $\bar{\Theta}_{est}$, the derivative of cost function has to be taken w.r.t. $\bar{\Theta}_{est}$ and is given as follows

$$\nabla_{\bar{\Theta}^T_{est}(t)} C\left(t\right) = -2e\left(t\right)\bar{X}\left(t\right) \ , \tag{17}$$

where the instantaneous estimation error $e(t)$ and the instantaneous estimate of the system output for corresponding instantaneous input are given by (15) and (16) respectively.

The update equation for LMS based Volterra model is derived based on the following mathematical expression

$$\bar{\Theta}^T_{est}(t+1) = \bar{\Theta}^T_{est}(t) - \frac{\alpha}{2}\nabla_{\bar{\Theta}^T_{est}(t)} C\left(t\right) \ . \tag{18}$$

Using equation (17) and (18), the final expression is derived for updating the instantaneous model parameters $\bar{\Theta}_{est}(t)$ and is given as

$$\bar{\Theta}^T_{est}(t+1) = \bar{\Theta}^T_{est}(t) + \alpha e\left(t\right)\bar{X}\left(t\right) \ , \tag{19}$$

where α is a small positive parameter which controls the convergence rate and steady state error of the Volterra model. When α is small convergence rate is slow, the steady state error is less and vice-versa. The value of α has to be bounded within certain limits so that model parameters does not grow unbounded and system remains stable. The bound on α for model stability is given as [15]

$$0 < \alpha < \frac{2}{\lambda_{\max}\left(R_X\right)} \ , \tag{20}$$

where $\lambda_{\max}$ is the largest eigenvalue of the covariance of the instantaneous input data $\bar{X}(t)$ i.e. R_X and is given as $R_X = \mathrm{E}\left[\bar{X}\left(t\right)\bar{X}(t)^T\right]$. The more robust bound on α value than (20) can be considered as

$$0 < \alpha < \frac{2}{Tr\left(R_X\right)} \ , \tag{21}$$

where $Tr\left(R_X\right)$ denotes the sum of all diagonal elements of R_X which is same as sum of all eigenvalues of R_X. If the value of α is not within this bounds, the stable performance of LMS based Volterra model parameter estimation can not be guaranteed. In LMS based Volterra model parameter estimation approach, we consider the statistics of input data to be known for deciding the appropriate value of learning parameter α.

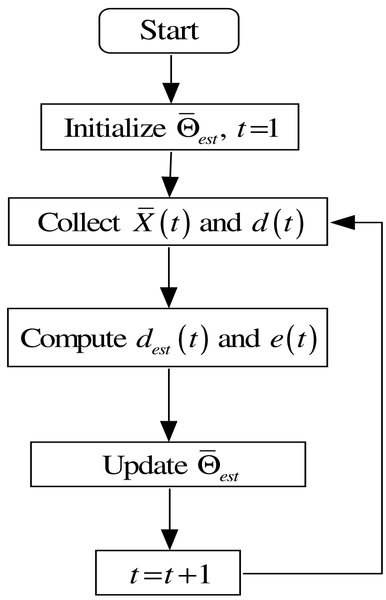

Fig. 4: Flowchart for estimation of Volterra model parameters using LMS algorithm

IV. Leaky LMS Based Volterra Model

The drawback of LMS based Volterra model parameter estimation approach is it's slow convergence rate and the input statistics are considered to be known to determine the appropriate value of α. When an unknown input signal is presented to the LMS based Volterra model parameter estimation approach, the algorithm behavior can not be predicted. Whenever the unknown input signal presented has very large eigenspread, the parameters in LMS algorithm grow unbounded and system becomes unstable. This behavior of LMS is known as weight drifting and this causes arithmetic overflow as well as unbounded growth of Volterra model parameters.

To mitigate this unbounded growth of parameters and to make LMS based Volterra model parameter estimation stable a small leakage factor is introduced in the cost function of LMS based Volterra model

$$C\left(t\right)=e(t)^2+\gamma\,\bar{\Theta}^T\left(t\right)\bar{\Theta}\left(t\right)\;, \tag{22}$$

where γ is known as leakage factor and lies in the range given as $0<\gamma<1$. Using basics from derivation of the LMS algorithm, the parameter update equation for leaky LMS based Volterra model becomes

$$\bar{\Theta}^T_{est}(t+1)=\left(1-\gamma\alpha\right)\bar{\Theta}^T_{est}(t)+\alpha e\left(t\right)\bar{X}\left(t\right) \tag{23}$$

and the bound on α for stability of algorithm becomes

$$\begin{aligned} &0<\alpha<\tfrac{2}{\gamma+\lambda_{\max}(R_X)} \\ &or\quad 0<\alpha<\tfrac{2}{\gamma+Tr(R_X)}\;. \end{aligned} \tag{24}$$

V. Modified Leaky LMS Based Volterra Model

The leaky LMS algorithm solves the problem associated with the stability of LMS based Volterra model parameter estimation in case of input signal with large eigenspread. But the convergence rate of both LMS and leaky LMS is still slow. To increase the convergence speed, a modified leaky LMS algorithm, the cost function has been modified by introducing the two exponentials in the cost function of leaky LMS [20]. The cost function of modified leaky LMS based Volterra model becomes

$$C\left(t\right)=\left(\exp\left(e\left(t\right)\right)+\exp\left(-e\left(t\right)\right)\right)^2+\gamma\,\bar{\Theta}^T_{est}\left(t\right)\bar{\Theta}_{est}\left(t\right) \tag{25}$$

and the instantaneous parameter update equation for Volterra model parameters becomes

$$\bar{\Theta}^T_{est}(t+1)=\left(1-\gamma\alpha\right)\bar{\Theta}^T_{est}(t)+2\alpha\bar{X}\left(t\right)\sinh\left(e\left(t\right)\right)\;, \tag{26}$$

here the term $\sinh\left(e\left(t\right)\right)$ increases the update value by large amount even for small value of instantaneous error. Thus increases the convergence rate significantly especially in the start. Also the leakage factor controls the unbounded growth of Volterra model parameters.

VI. Simulation Example

A. Controlling the unbounded nature of LMS based Volterra model:

Simulation setup: The system with following nonlinear input-output relation has been considered for simulation

$$\begin{aligned} d\left(t\right)=&\,2.73\,x(t)+0.5\,x(t-1)+1.12\,x(t-2)+0.9\,x(t)^2\\ &+1.35\,x(t-1)^2-2.8\,x(t-2)^2+0.39\,x(t)\,x(t-1)\\ &-1.35\,x(t-1)\,x(t-2)-0.67\,x(t)x(t-2)\;, \end{aligned} \tag{27}$$

where $d\left(t\right)$ is the instantaneous system output, $x\left(t\right)$ is the instantaneous input and $x\left(t-1\right)$, $x\left(t-2\right)$ are previous inputs.

According to equations (8), (9), (11) and (12), the second-order Volterra model parameters for a system with fading memory factor $M=3$ are given as

$$\bar{\Theta}\;=\;\left[\bar{h}_1,\bar{h}_2\right]^T\in\mathbb{R}^{12}\;, \tag{28}$$

where

$$\bar{h}_1=\left[h_1\left(1\right),...,h_1\left(3\right)\right]\in\mathbb{R}^3\;, \tag{29}$$

$$\bar{h}_2=\left[h_2\left(1\right),...,h_2\left(9\right)\right]\in\mathbb{R}^9 \tag{30}$$

and the instantaneous input sequence $X(t)$ is given as

$$\bar{X}\left(t\right)=\left[\begin{array}{l} x\left(t\right),\;x\left(t-1\right),\;x\left(t-2\right),\;x^2\left(t\right),\\ x^2\left(t-1\right)x^2\left(t-2\right),\;x\left(t\right)x\left(t-1\right),\\ x\left(t-1\right)x\left(t-2\right),\;x\left(t\right)x\left(t-2\right) \end{array}\right]^T\;. \tag{31}$$

Due to symmetry property of volterra kernels the parameters in $\bar{h}_2$ are reduced to six and hence the total parameters in $\bar{\Theta}$ are effectively nine. The $\bar{\Theta}_{opt}$ in (27) for above example is given as

$$\bar{\Theta}_{opt}=\left[\begin{array}{l} 2.73,\;0.5,\;1.12,\;0.9,\;1.35,\\ -2.8,\;0.39,\;-1.35,\;-0.67 \end{array}\right]\;. \tag{32}$$

To analyze the performance of LMS, leaky LMS and modified leaky LMS based Volterra model, the metrics mean-square deviation (MSD) and mean-square error (MSE) are considered. The parameters MSD and MSE at any time instant t are defined as

$$MSD\left(t\right)=E\|\bar{\Theta}_{est}(t)-\bar{\Theta}_{opt}(t)\|^2 \tag{33}$$

$$MSE\left(t\right)=E\|b_{est}(t)-b(t)\|^2 \tag{34}$$

The model is trained for random Gaussian input signals with eigenspread less than 0.2×10^3 and hence the stability bound on α becomes $0<\alpha<0.02$ (from 20). Learning parameter α is taken as 0.01 and product of α and γ to be 0.5×10^{-4}. The simulation is performed on 10,000 input samples and the performance of LMS and leaky LMS under normal operating condition is given in Fig. 5.

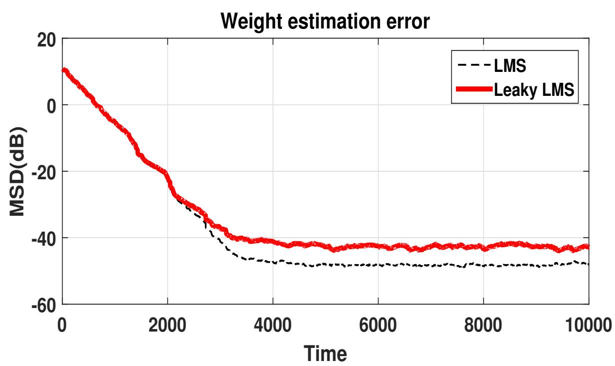

Fig. 5: Mean-square deviation for input with small eigenspread

But when the random Gaussian signal with eigenspread

0.5×10^3 is presented to the LMS based Volterra model, the parameters grow unbounded and may result in arithmetic overflow and affect the system stability as well.

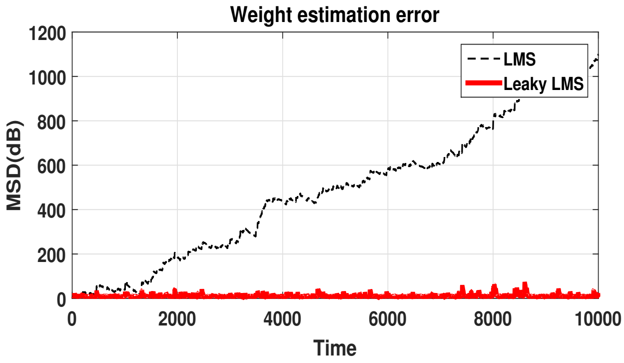

Fig. 6: Mean-square deviation for input with large eigenspread

It can be seen from Fig. 6, the leaky LMS based Volterra model limits the unbounded parameter growth and avoids arithmetic overflow. This is achieved with little degradation in performance of the model as can be seen from Fig. 5. The selection of leakage factor is a critical issue. The leakage factor has to be selected based on performance requirements of the different applications.

The leaky LMS based approach did not solve slow convergence problem in traditional LMS based Volterra model as it can be seen from Fig. 5. To improve the convergence rate, modified leaky LMS based Volterra model is considered and simulation results are presented in the following subsection.

B. Improving the convergence rate of LMS and leaky LMS based Volterra model:

The same system which is considered for above simulation is considered for this simulation. The input signals are taken as random Gaussian signal having variance 1 and the random Gaussian noise with variance 10^{-3} is added to the signal. The α value is taken as 0.01 and the product of α and γ is taken to be 0.5×10^{-4}. The above example is simulated over 10000 samples i.e. $T = 10,000$ and the performance parameters are plotted in the Fig. 7.

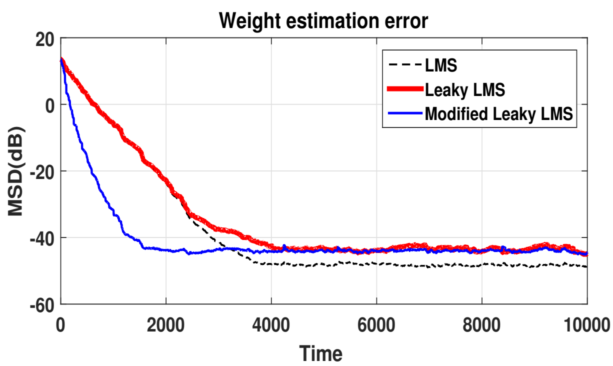

Fig. 7: Mean-square deviation in Volterra model parameter estimation

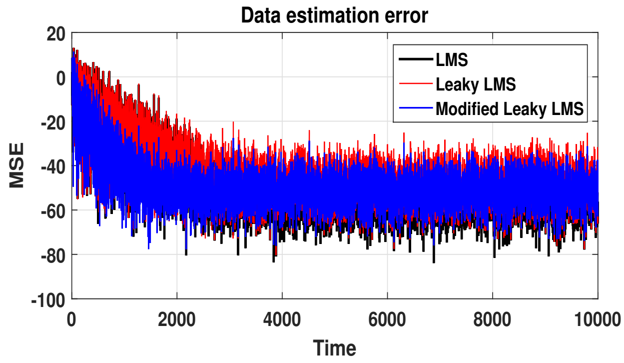

Fig. 8: Mean-square error in system output prediction

It can be seen from Fig. 7, the modified leaky LMS based Volterra model parameter estimation improves the convergence rate as compared to LMS and leaky LMS based Volterra model parameter estimation. But the performance of modified leaky is slightly lower compared to LMS based Volterra model. In Fig. 9, the predictions of $1^{st}-$order and $2^{nd}-$order Volterra model approximations are plotted to show it's effectiveness in modeling nonlinear dynamical systems.

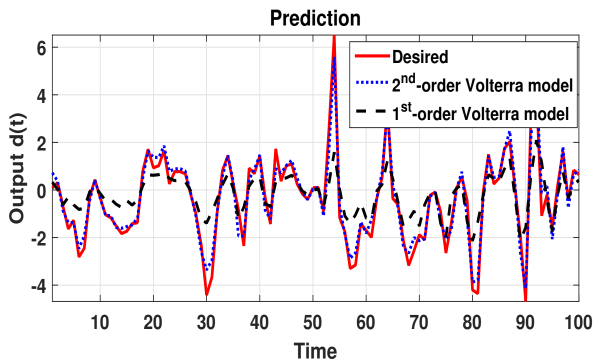

Fig. 9: Prediction performance of $1^{st}-$order and $2^{nd}-$order Volterra model approximations

VII. Conclusion

This article demonstrated the Volterra model parameter estimation approaches based on LMS algorithm and it's two variants. The LMS based Volterra model parameter estimation approach has slow convergence rate and unstable behavior. The MATLAB simulations presented in this work show that the leaky LMS and modified leaky LMS based methodologies are very effective in solving these problems. The model can be made more accurate and stable by appropriate selection of the leakage factor.

References

[1] S. Billings, "Identification of nonlinear systems–a survey," in *IEE Proceedings D (Control Theory and Applications)*, vol. 127, no. 6. IET, 1980, pp. 272–285.

[2] M. Enqvist, "Linear models of nonlinear systems," 2005.

[3] L. Ljung, "Approaches to identification of nonlinear systems," in *Control Conference (CCC), 2010 29th Chinese*. IEEE, 2010, pp. 1–5.

[4] A. Tavakolpour-Saleh, S. Nasib, A. Sepasyan, and S. Hashemi, "Parametric and nonparametric system identification of an experimental turbojet engine," *Aerospace Science and Technology*, vol. 43, pp. 21–29, 2015.

[5] A. Stenger and R. Rabenstein, "Adaptive volterra filters for nonlinear acoustic echo cancellation." in *NSIP*, 1999, pp. 679–683.

[6] A. Khan and N. Vyas, "Application of volterra and wiener theories for nonlinear parameter estimation in a rotor-bearing system," *Nonlinear Dynamics*, vol. 24, no. 3, pp. 285–304, 2001.

[7] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive volterra filters," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 877–880.

[8] D. FJ III, R. K. Pearson, and B. A. Ogunnaike, *Identification and control using Volterra models*. Springer Science & Business Media, 2012.

[9] W. Suleiman and A. Monin, "New method for identifying finite degree volterra series," *Automatica*, vol. 44, no. 2, pp. 488–497, 2008.

[10] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[11] T. Ogunfunmi, "Nonlinear adaptive system identification based on volterra models," *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*, pp. 115–128, 2007.

[12] D. I. Soloway and J. T. Bialasiewicz, "Neural network modeling of nonlinear systems based on volterra series extension of a linear model," in *Intelligent Control, 1992., Proceedings of the 1992 IEEE International Symposium on*. IEEE, 1992, pp. 7–12.

[13] X. Wen and D. Luo, "Application of volterra LMS adaptive filter algorithm based on gaussian distribution," 2013.

[14] G. Budura and C. Botoca, "Nonlinearities identification using the LMS volterra filter," *Communications Department Faculty of Electronics and Telecommunications Timisoara, Bd. V. Parvan*, no. 2, 2005.

[15] D. Bismor and M. Pawelczyk, "Stability conditions for the leaky LMS algorithm based on control theory analysis," *Archives of Acoustics*, vol. 41, no. 4, pp. 731–739, 2016.

[16] M. A. Nasar and A. Zerguine, "The leaky least mean mixed norm algorithm," in *Signals, Systems and Computers, 2013 Asilomar Conference on*. IEEE, 2013, pp. 1520–1523.

[17] M. Sowjanya, A. Sahoo, and S. Kumar, "Distributed incremental leaky LMS," in *Communications and Signal Processing (ICCSP), 2015 International Conference on*. IEEE, 2015, pp. 1753–1757.

[18] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Parameter estimation of wiener nonlinear model using least mean square (LMS) algorithm." IEEE Region Ten Conference (TENCON-2017), Penang, Malaysia, 5-8 November, 2017, 2017.

[19] C. Cheng, Z. Peng, W. Zhang, and G. Meng, "Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review," *Mechanical Systems and Signal Processing*, vol. 87, pp. 340–364, 2017.

[20] T. R. Gwadabe, M. S. Salman, and H. Abuhilal, "A modified leaky-LMS algorithm," *International Journal of Computer and Electrical Engineering*, vol. 6, no. 3, p. 222, 2014.